2026/08/07, 2026 네트워크 기초 세미나

TCP/IP 프로토콜

-4장 네트워크 계층의 소개, 5장 IPv4 주소-

서 진 우(jinwoo@pel.sejong.ac.kr)

세종대학교 프로토콜공학연구실

목 차

- 네트워크 계층의 소개
- IPv4 주소

목 차

네트워크 계층의 소개

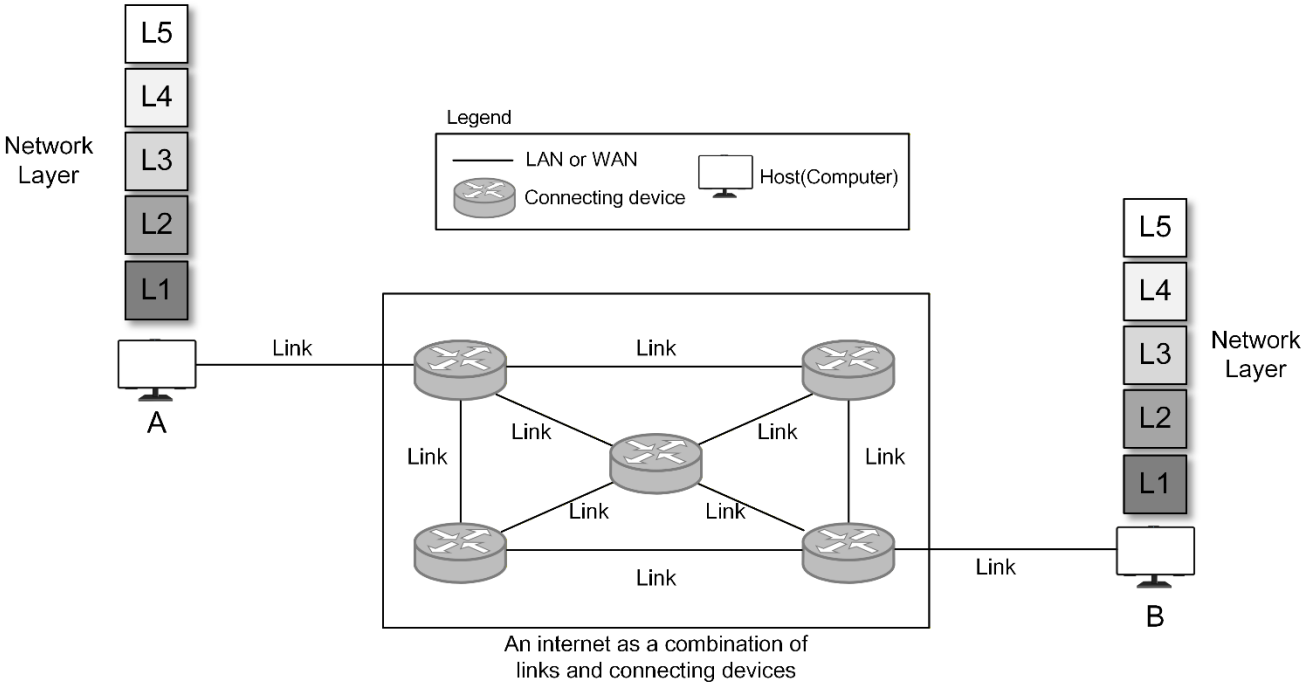
- 네트워크 계층 (1/2)
  - 정의
    - OSI 모델의 3계층으로, 데이터 전송을 위한 경로 선택 및 패킷 전달을 담당하는 계층
  - 주요 개념 (1/2)
    - 라우터(Router)
      - 서로 다른 네트워크를 연결하여 패킷이 빠르고 안전하게 도달할 수 있도록 최적의 경로를 지정하는 네트워크 장비
    - 라우팅(Routing)
      - 데이터가 출발지로부터 목적지까지 갈 수 있는 최적의 경로를 설정하는 과정
        - e.g., 정적 라우팅, 동적 라우팅
    - 포워딩(Forwarding)
      - 결정된 경로를 따라 패킷을 실제로 다음 목적지로 전달하는 동작

*정적 라우팅(Static Routing)
괸리자가 라우팅 정보를 수동으로 직접 설정하는 방식.
*동적 라우팅(Dynamic Routing)
라우팅 프로토콜을 이용해 라우터들이 자동으로 경로 정보를 교환하고 갱신하는 방식.

네트워크 계층의 소개

- 네트워크 계층 (2/2)
  - 주요 개념 (2/2)
    - 인터네트워크(InterNetwork)
      - LAN과 WAN의 조합으로 구성된 네트워크
    - 인터넷(Internet)
      - 전 세계적으로 연결된 특정한 하나의 인터네트워크

*LAN (Local Area Network)
제한된 영역 내에서 기기를 상호 연결하는 근거리 네트워크.
*WAN (World Area Network)
멀리 떨어진 LAN과 LAN을 연결하는 국가나 대륙 규모의 대규모 네트워크.



네트워크 계층의 소개

- 교환(Switching) (1/3)
  - 정의
    - 데이터를 전달하는 방식
  - 종류(1/2)
    - 회선 교환(Circuit Switching)
      - 정의
        - 메시지 전달 전 발신지와 목적지 사이에 물리적 회선을 미리 설정하는 교환 방식
      - 특징
        - 회선 설정 후 전체 메시지 전송
        - 전송 완료 시 발신지가 네트워크에 통보하여 회선을 해제
        - 회선 독점을 통한 통신 방식
          - e.g., 음성 전화 시스템

네트워크 계층의 소개

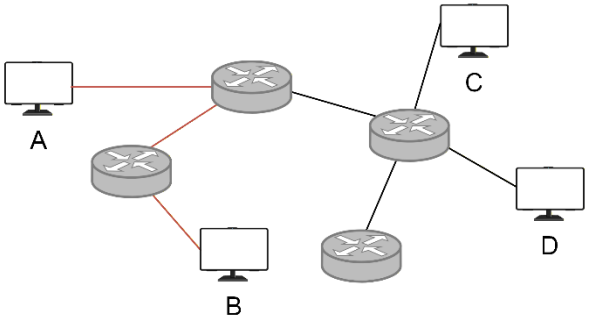
- 교환(Switching) (2/3)
  - 종류 (2/2)
    - 패킷 교환(Packet Switching)
      - 정의
        - 상위 계층의 메시지를 관리 가능한 크기의 패킷으로 분할하여 전달하는 교환 방식
      - 특징
        - 발신지는 패킷을 하나씩 전달하며 목적지는 모든 패킷 도착을 기다린 후 상위 계층에 전달
          - e.g., 카카오톡, 이메일
        - 패킷들은 서로 다른 경로로의 전송이 가능하여 송/수신의 패킷 순서의 불일치 가능성 존재
        - 데이터그램 방식과 가상 회선 방식으로 세분화

*데이터그램(Datagram)
패킷 교환 네트워크와 관련된 기본 전송 단위.
*데이터그램 방식
비연결형 서비스에서 사용, 각 패킷의 독립적 경로 결정 및 전달.
*가상 회선 방식
연결형 서비스에서 사용, 전송 전 경로 설정 후 전체 패킷의 동일 경로 사용.

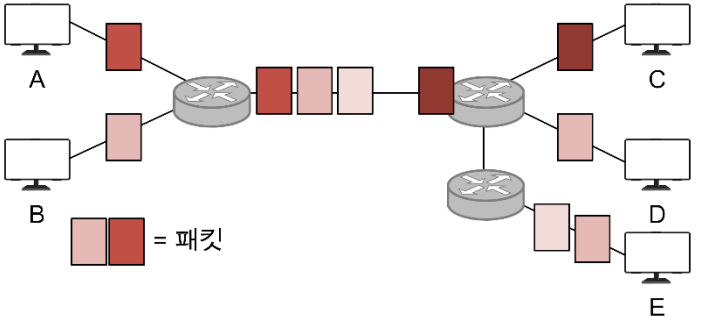
네트워크 계층의 소개

- 교환(Switching) (3/3)
  - 장단점

	회선 교환 (Circuit Switching)	패킷 교환 (Packet Switching)
장점	대역폭 보장으로 안정적인 전송 가능	자원 공유로 높은 효율성
	자동적으로 순서 보장	장애 발생 시 경로 우회로 높은 유연성
	지연 예측 가능	다수 동시 통신 수용으로 높은 확장성
단점	설정 초기 지연 발생	도착 순서의 역전 가능성
	미사용 회선 독점으로 자원 낭비	패킷별 헤더로 인한 오버헤드
	낮은 확장성	트래픽에 따른 지연 편차 발생



회선 교환



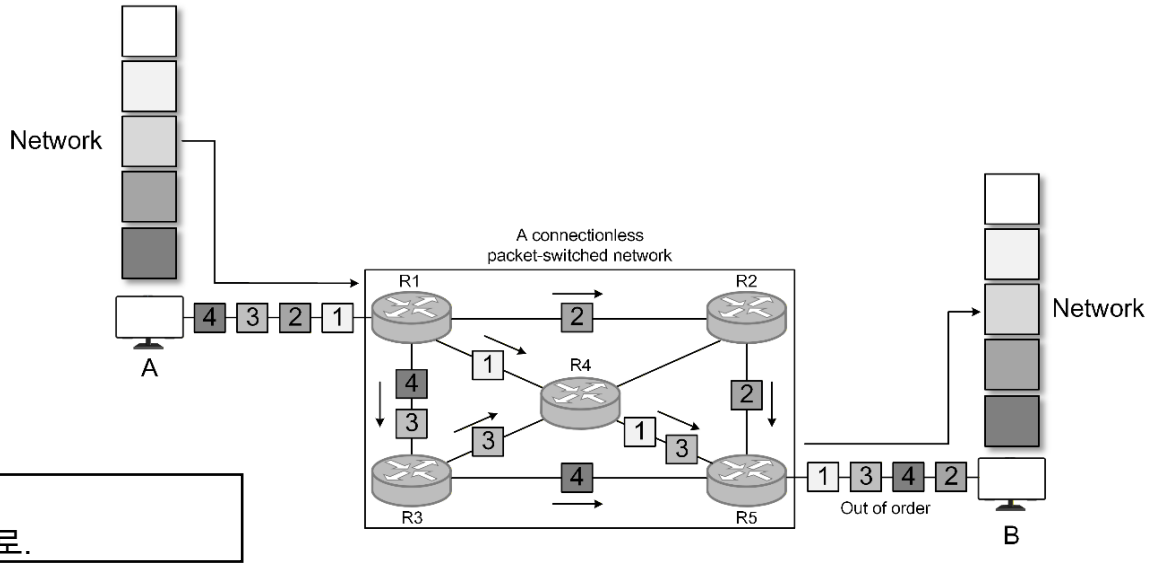
패킷 교환

네트워크 계층의 소개

- 네트워크 계층에서의 패킷 교환
  - 특징
    - 네트워크 계층은 패킷 교환 네트워크로 설계
      - 메시지를 데이터그램 단위로 분할하여 전송
    - 목적지에서 데이터그램을 재조립하여 원본 메시지 전달
  - 종류
    - 비연결형 서비스(Connectionless Service)
    - 연결형 서비스(Connection-oriented Service)

네트워크 계층의 소개

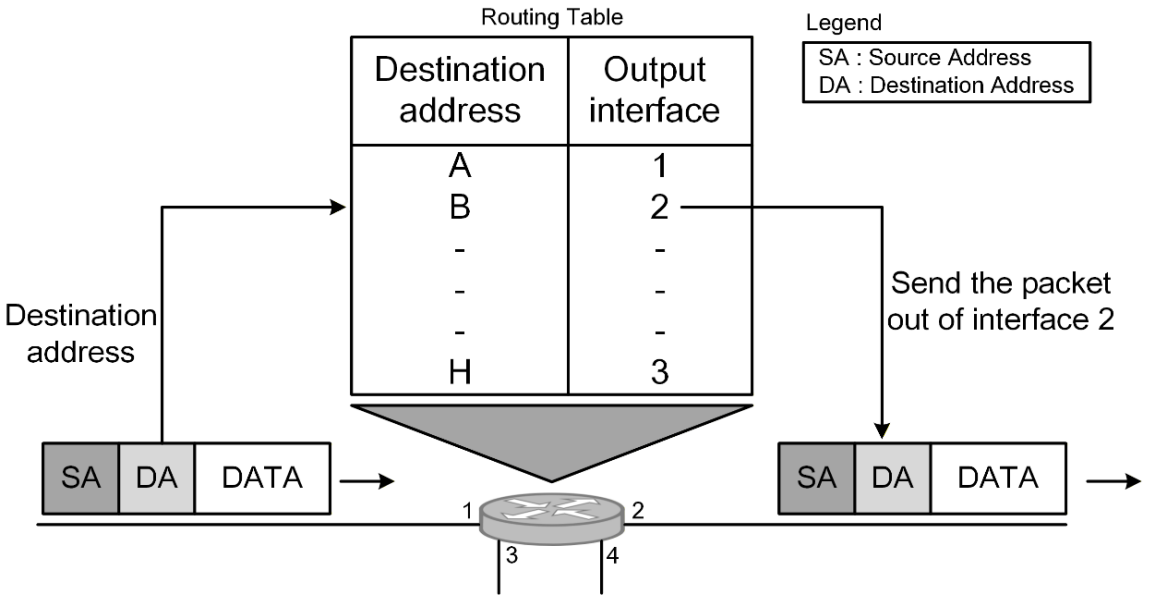
- 비연결형 서비스(Connectionless Service) (1/3)
  - 정의
    - 송신자와 수신자 사이에 연결을 확립하지 않고 데이터를 전송하는 방법
  - 특징
    - 각 패킷을 상호독립적 취급하여 독립적 경로 선택 가능
    - 신뢰성 없는 전송
  - 예시
    - IP (Internet Protocol)



*IP (Internet Protocol)
인터넷 프로토콜 스위트에서 네트워크 계층의 통신 프로토콜로.

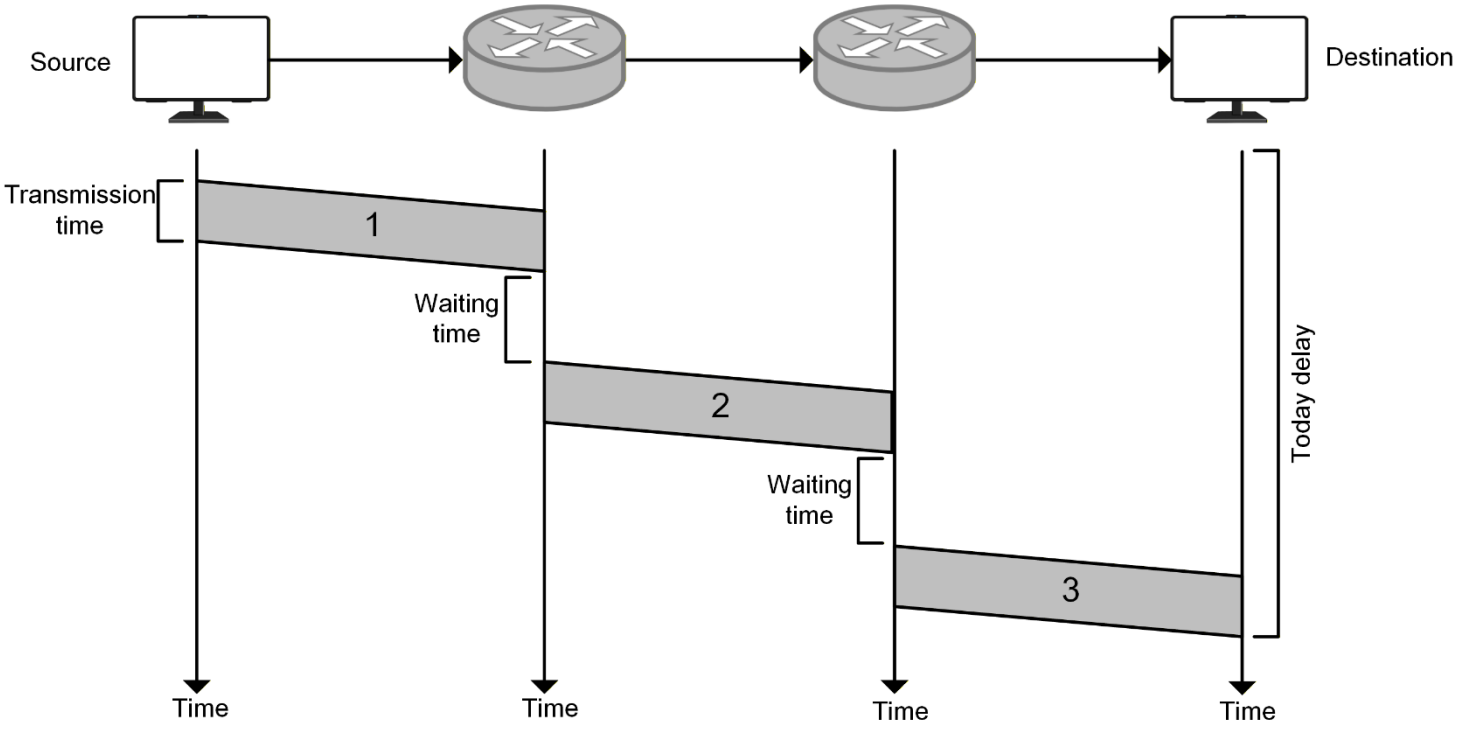
네트워크 계층의 소개

- 비연결형 서비스(Connectionless Service) (2/3)
  - 포워딩 과정
    - 각 패킷은 헤더에 속한 정보인 발신지 주소와 목적지 주소를 기반으로 전달
    - 라우터는 목적지 주소에 기반하여 패킷을 전달
    - 패킷이 폐기되는 경우 발신지 주소를 사용하여 발신지에 ICMP 오류 메시지를 보내는 것이 가능



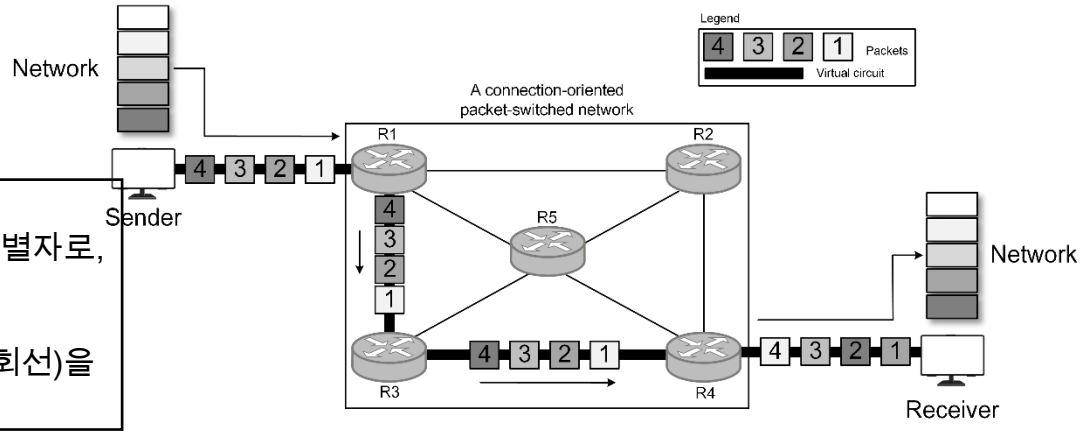
네트워크 계층의 소개

- 비연결형 서비스(Connectionless Service) (3/3)
  - 비연결형 네트워크에서의 지연
    - 패킷이 분실되거나 재전송되는 경우
    - 목적지에서 모든 패킷을 수신할 때까지 기다려야 하는 경우



네트워크 계층의 소개

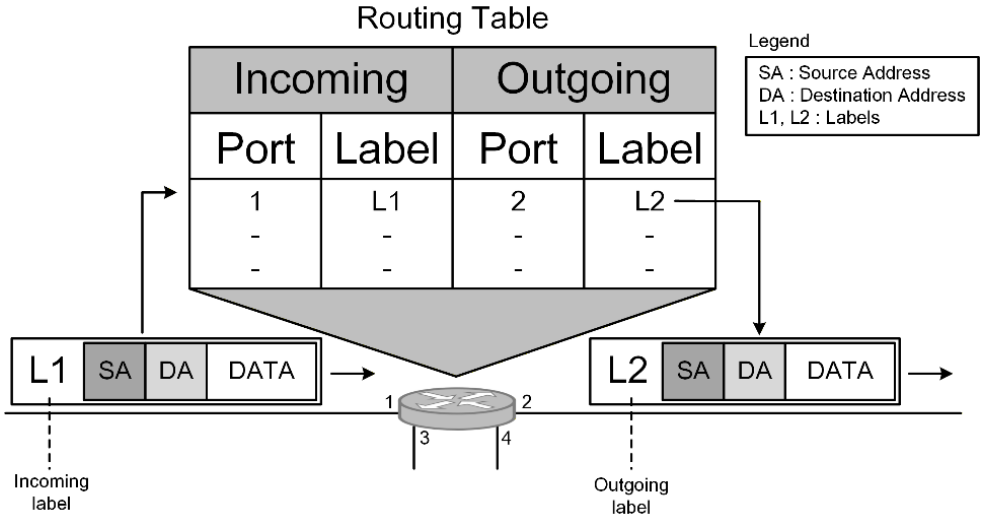
- 연결형 서비스(Connection-oriented Service) (1/6)
  - 정의
    - 송신자와 수신자 사이의 논리적인 연결을 확립하고 데이터를 전송하는 방법
  - 특징
    - 데이터그램 전송 전, 경로를 정의하기 위한 가상 회선 (Virtual Circuit) 설정 필요
    - 회선 생성 후 데이터그램이 동일한 경로 추종
    - 패킷 내 가상 회선 식별자인 흐름 레이블(Flow Label) 존재
  - 예시
    - X.25



*흐름 레이블(Flow Label)
연결형 서비스(가상 회선 방식)에서 각 패킷이 헤더에 담고 다니는 식별자로, 해당 패킷이 어느 가상 회선을 따라가야 하는지를 나타냄.
*X.25
데이터를 보내기 전에 송신측과 수신측 사이에 전용 데이터 길(가상 회선)을 먼저 연결해 두고 데이터를 주고받는 방식.

네트워크 계층의 소개

- 연결형 서비스(Connection-oriented Service) (2/6)
  - 포워딩
    - 특징
      - 각 패킷은 흐름 레이블(Flow Label)을 기준으로 포워딩
    - 과정
      1. 설정 과정(Setup Phase)
      2. 데이터 전송 과정(Data Transfer Phase)
      3. 해체 과정(Teardown Phase)

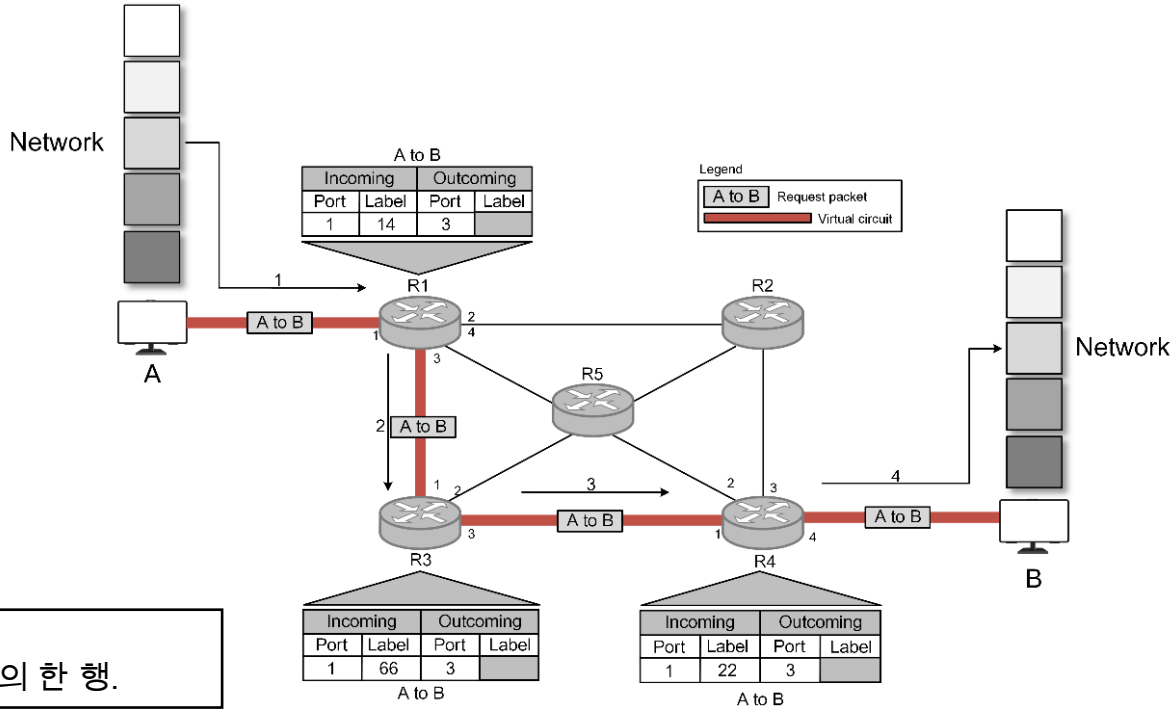


네트워크 계층의 소개

- 연결형 서비스(Connection-oriented Service) (3/6)
  - 포워딩 과정 (1/3)
    1. 설정 단계(Setup Phase) (1/2)
       - 라우터가 가상회선을 위한 엔트리(Entry)를 생성하는 단계
       - 발신지 A에서 목적지 B로 가상회선을 만들려는 경우
         - 요청 패킷(Request Packet)과 확인응답 패킷(Acknowledgement Packet) 교환

1. 요청 패킷 단계

- 출발지에서 목적지까지 경로상 각 라우터가 입출력 포트와 입력 레이블을 순차적으로 결정하며 가상 회선 설정



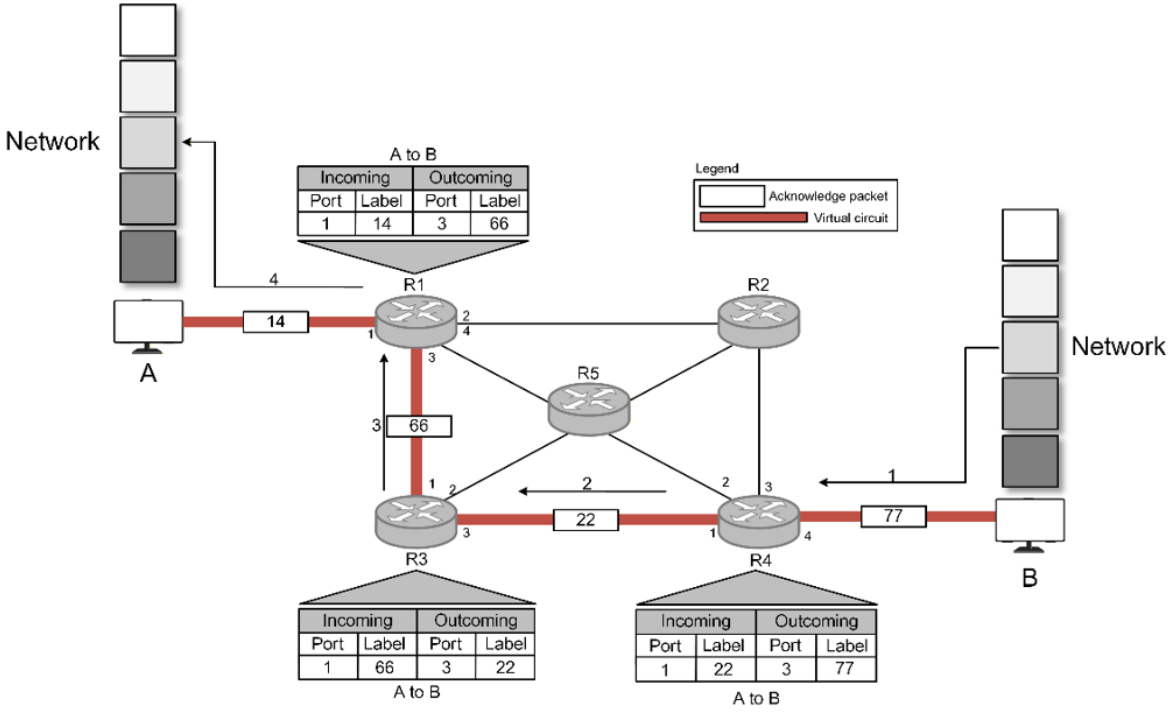
*엔트리(Entry)
연결 설정 단계에서 경로상의 각 라우터가 만들어두는 테이블의 한 행.

네트워크 계층의 소개

- 연결형 서비스(Connection-oriented Service) (4/6)
  - 포워딩 과정 (2/3)
    1. 설정 단계(Setup Phase) (2/2)
       - 라우터가 가상회선을 위한 엔트리(Entry)를 생성하는 단계
       - 발신지 A에서 목적지 B로 가상회선을 만들려는 경우
         - 요청 패킷(Request Packet)과 확인응답 패킷(Acknowledgement Packet) 교환

2. 확인응답 패킷 단계

- 목적지에서 발신지 방향으로 역순 전파하며 각 라우터의 출력 레이블을 완성



네트워크 계층의 소개

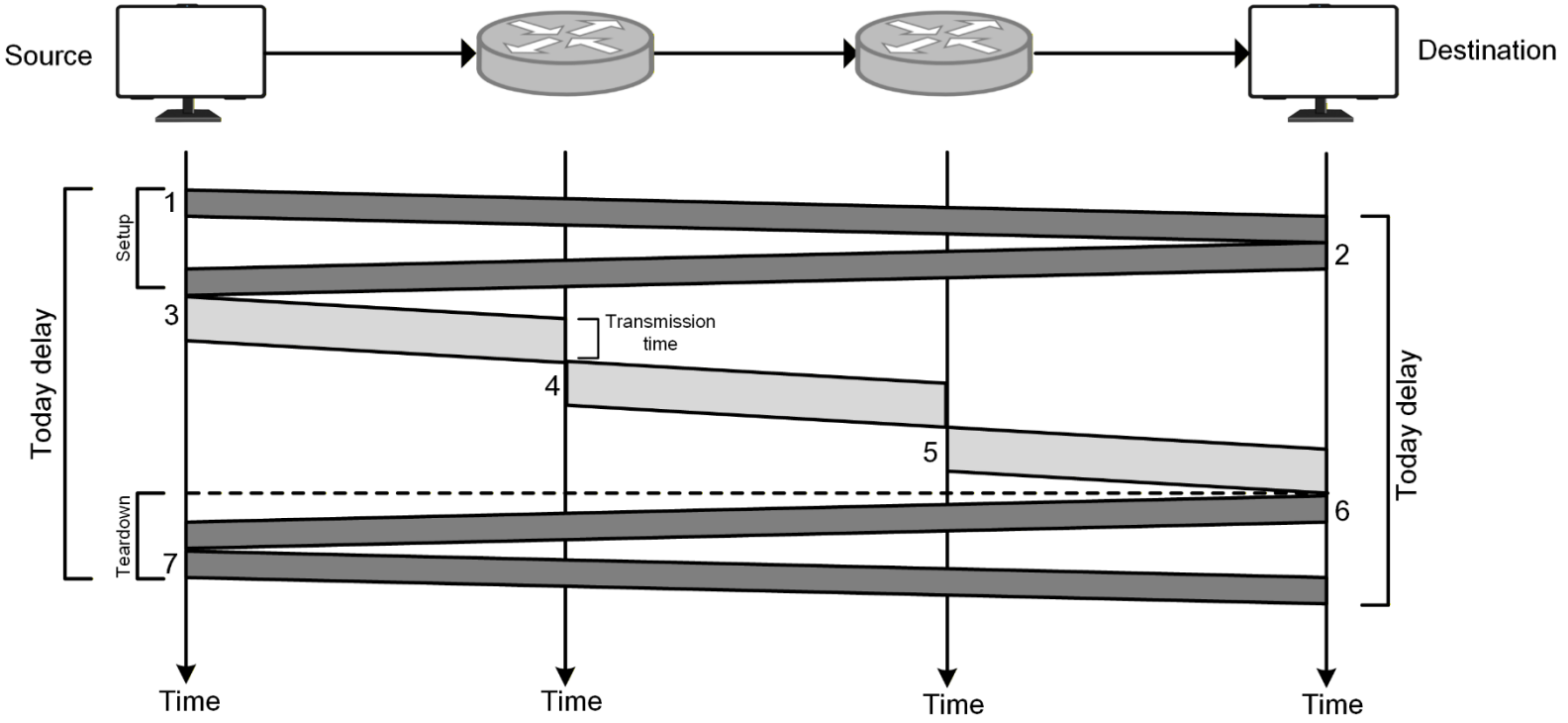
- 연결형 서비스(Connection-oriented Service) (5/6)
  - 포워딩 과정 (3/3)

    2. 데이터 전송 과정(Data Transfer Phase)
       - 가상 회선 설정 완료 후, 모든 패킷이 동일한 레이블 순서를 따라 목적지까지 순차 전달
       - 패킷 수와 무관하게 동일한 경로 및 레이블 순서 적용

    3. 해체 과정(Teardown Phase)
       - 전송 완료 후 발신지가 해체 패킷을 목적지로 전송
       - 목적지는 확인 패킷(Confirmation Packet)으로 응답
       - 모든 라우터들은 자신의 테이블에서 해당 엔트리 삭제

네트워크 계층의 소개

- 연결형 서비스(Connection-oriented Service) (6/6)
  - 연결형 서비스에서의 지연
    - 연결을 확립하는 과정에서 지연이 발생하는 경우
    - 패킷 분실로 인해 재전송 과정에서 지연이 발생하는 경우
    - 재전송 절차를 무시하여 지연이 발생하는 경우



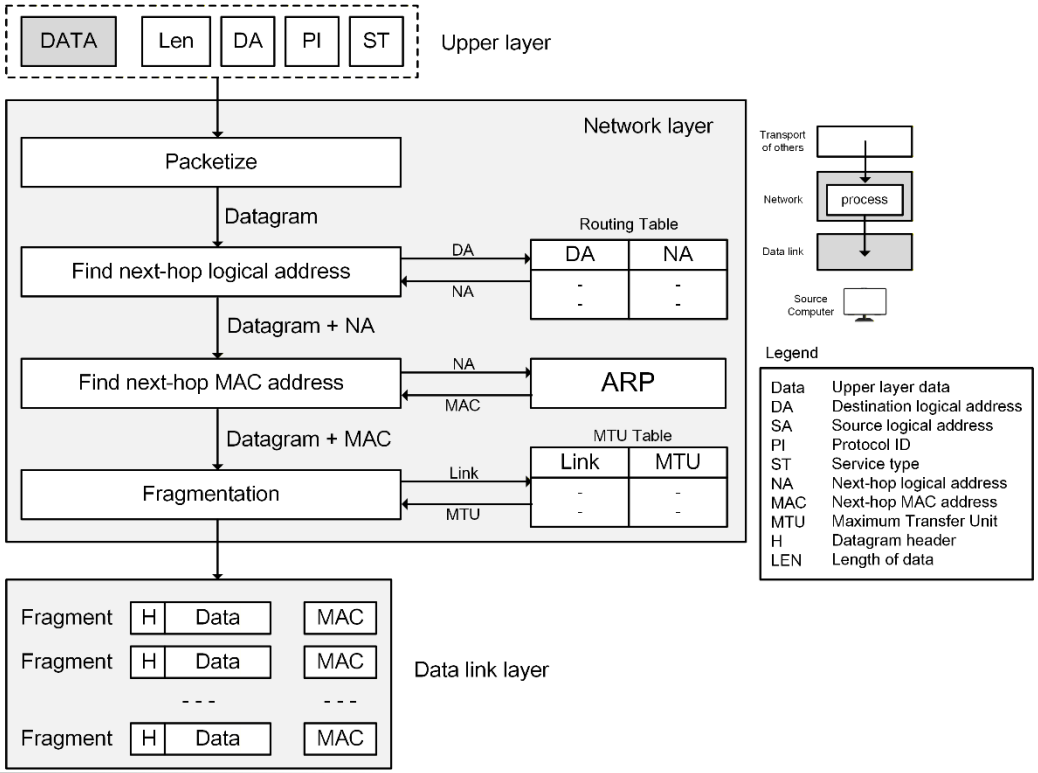
네트워크 계층의 소개

- 네트워크 계층 서비스 (1/5)
  - 논리 주소 체계
    - 종단 간(End-to-End) 통신 지원
      - 서로 다른 네트워크에 있는 두 컴퓨터가 서로를 식별하고 통신
      - 전 세계의 모든 컴퓨터에 고유한 네트워크 계층 주소(논리 주소)를 부여
    - 글로벌 주소 지정 메커니즘
      - 하위 계층(데이터 링크 계층)의 MAC 주소는 제조사마다 다르고 파편화되어 있지만, 네트워크 계층은 전 세계 어디서나 통용되는 균일하고 체계적인 주소(IPv4, IPv6)를 제공

*IPv4 (Internet Protocol version 4)
네트워크에서 호스트 간의 데이터 전송을 위해 경로 및 목적지를 지정하는 32비트 체계의 고유한 주소.
*IPv6 (Internet Protocol version 6)
IPv4의 32비트 주소 고갈 문제를 해결하기 위해 등장한, 128비트 체계의 고유한 주소를 사용하는 차세대 인터넷 프로토콜.

네트워크 계층의 소개

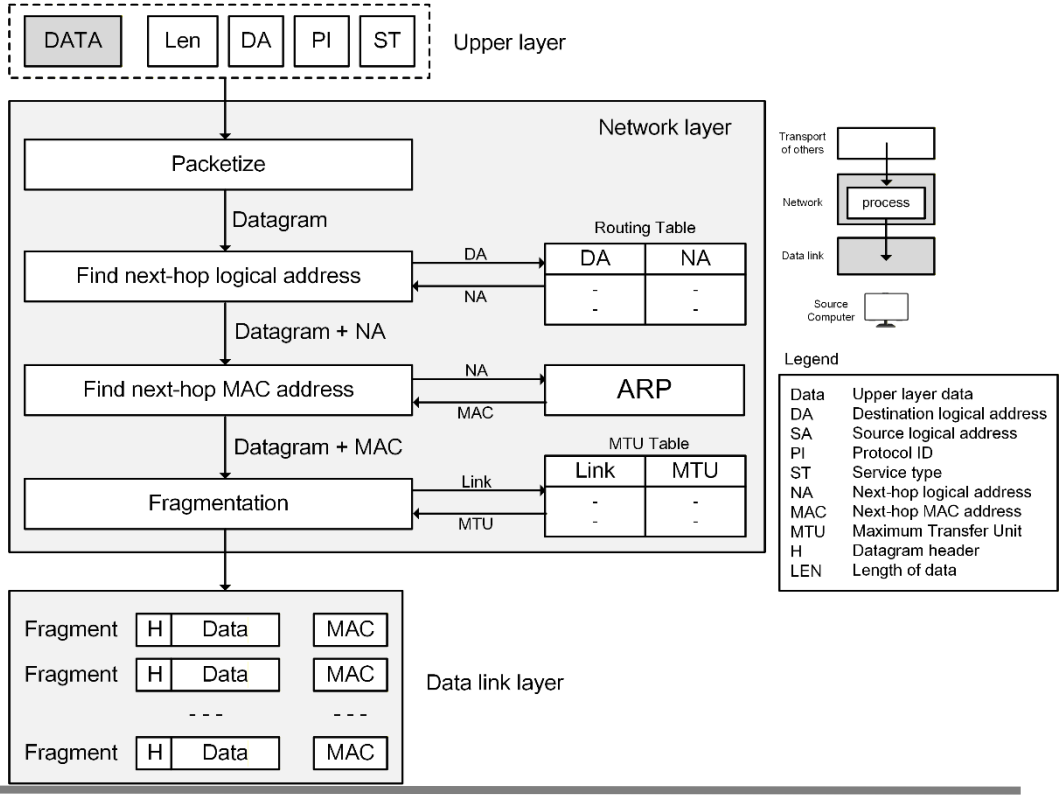
- 네트워크 계층 서비스 (2/5)
  - 발신지 컴퓨터에서 제공되는 서비스 (1/2)
    - 패킷화(Packetizing)
      - 상위 계층 데이터를 데이터그램으로 캡슐화
      - 헤더에 발신지·목적지 주소, 단편화 정보, 프로토콜 ID, 데이터 길이, checksum 등 포함
    - 다음 홉의 논리 주소 탐색
      - 같은 네트워크인 경우
        - 목적지의 IP 주소
      - 다른 네트워크인 경우
        - 라우팅 테이블을 조회해 다음 라우터의 IP 주소 탐색



*체크섬(Checksum)
데이터나 파일이 전송, 저장되는 과정에서 손상되거나 변조되지 않았는지 확인하기 위해 사용되는 데이터 무결성 검증 값.
*홉(Hop)
패킷이 현재 라우터에서 다음으로 이동해야 할 인접 라우터나 목적지를 의미.

네트워크 계층의 소개

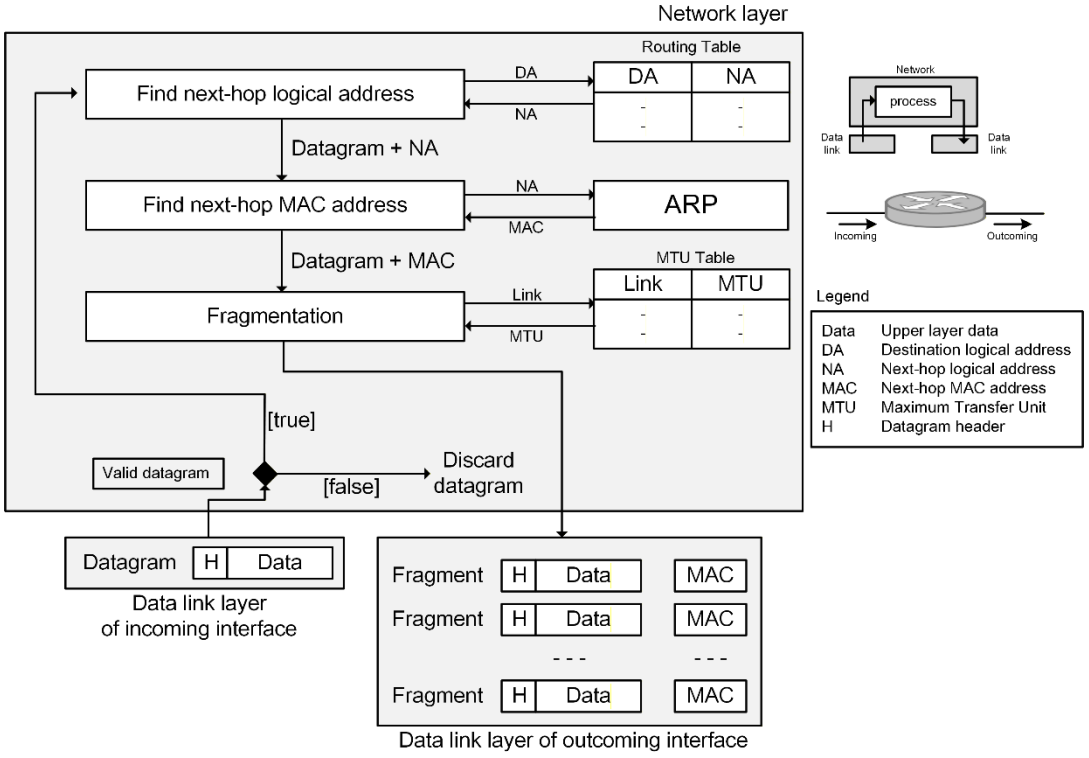
- 네트워크 계층 서비스 (3/5)
  - 발신지 컴퓨터에서 제공되는 서비스 (2/2)
    - 다음 홉의 물리(MAC) 주소 탐색
      - 실제 전달은 데이터링크 계층이 담당하며 MAC 주소 필요
      - ARP(Address Resolution Protocol)가 IP를 받아 MAC 주소로 매핑
    - 단편화(Fragmentation)
      - 데이터그램이 MTU (Maximum Transmission Unit)보다 크면 단편화
      - 각 단편에 헤더 반복
      - 각 단편에 위치 정보 추가



*MAC (Media Access Control Address)
네트워크 기기에 부여된 물리적인 고유 식별 번호.
*ARP (Address Resolution Protocol)
논리적인 IP 주소를 물리적인 MAC 주소로 변환해주는 통신 프로토콜.
*MTU (Maximum Transmission Unit)
네트워크에 연결된 장치가 받아들일 수 있는 최대 데이터 크기.

네트워크 계층의 소개

- 네트워크 계층 서비스 (4/5)
  - 각 라우터에서 제공되는 서비스
    - 입력·출력 인터페이스 상호작용
      - 라우터는 입력/출력 데이터링크 계층과 상호작용
    - 데이터그램 처리
      - 입력에서 데이터그램 수신
      - 필요시 단편화 수행
      - 출력으로 전달
    - 네트워크 계층 기능 수행
      - 다음 홉 논리 주소 탐색
      - 다음 홉 MAC 주소 탐색
      - 단편화 기능 지원
    - checksum 검증

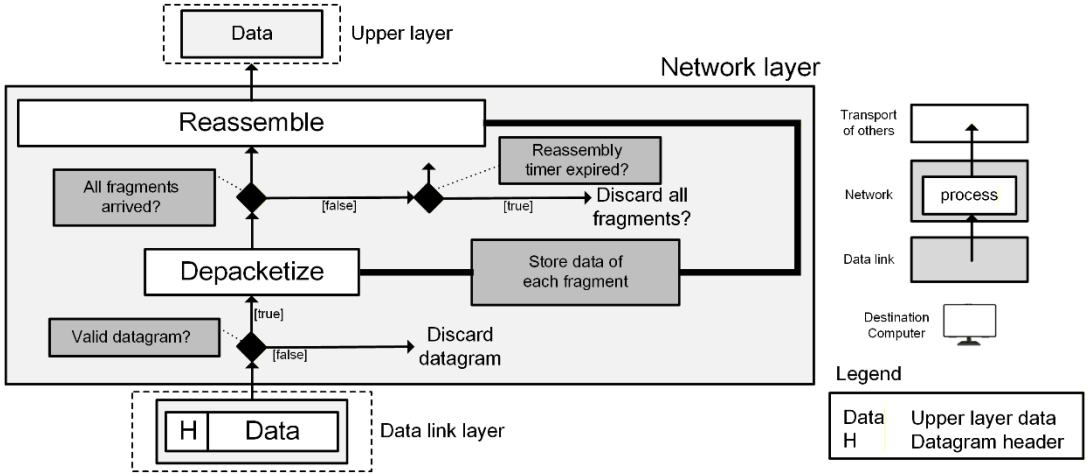


네트워크 계층의 소개

- 네트워크 계층 서비스 (5/5)

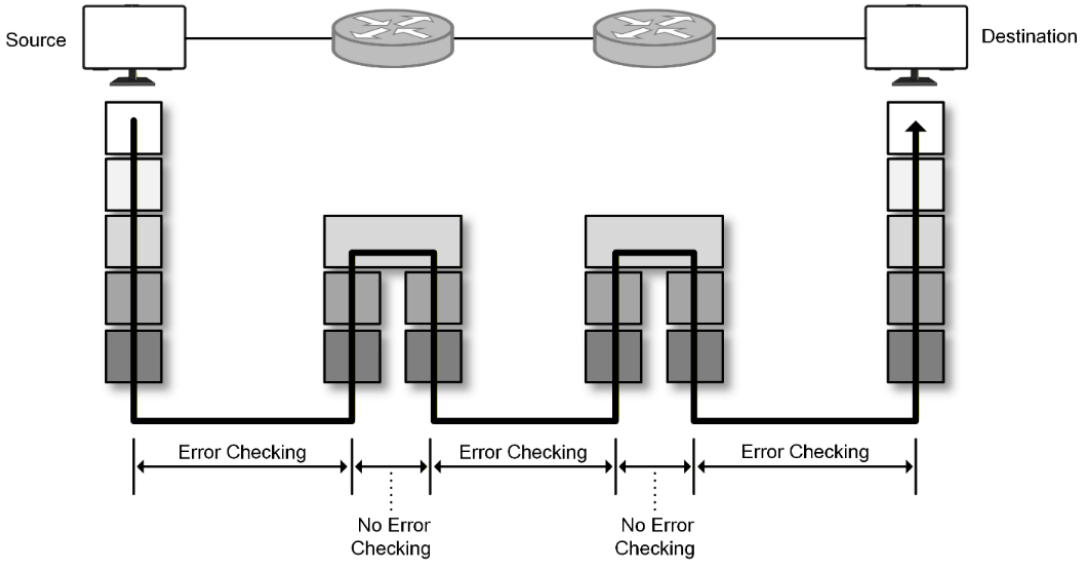
*ICMP (Internet Control Message Protocol)
네트워크 기기들이 통신 문제 진단 및 오류 보고를 위해 사용하는 네트워크 계층 프로토콜.

  - 목적지 컴퓨터에서 제공되는 서비스
    - 단편 도착 시 타이머 설정
      - 첫 번째 단편이 도착하면 재조립 타이머를 가동
    - 도착한 단편의 체크섬 검증
      - 수신한 각 단편의 IP 헤더 체크섬을 계산하여 데이터 유효성을 검증
    - 타이머 만료 시 폐기 및 오류 알림
      - 타이머가 만료될 때까지 모든 단편이 오지 않으면, 기존 단편을 모두 폐기하고 송신 측에 ICMP 오류 메시지를 전달
    - 모든 단편 도착 후 재조립 및 상위 계층 전달



네트워크 계층의 소개

- 네트워크 계층 문제점
  - 인터넷의 네트워크 계층은 비연결형 서비스 기반
  - 종류 (1/4)
    - 오류 수정 불가능
      - checksum으로 오류 탐지는 가능하나 수정 기능은 부재
      - 라우터 처리 중 발생하는 오류로 인한 잘못된 데이터 전달 가능성



네트워크 계층의 소개

- 네트워크 계층 문제점
  - 종류 (2/4)
    - 흐름 제어 부재
      - 오류 제어 기능 부재
        - 오류 제어 기능이 없어 수신 측 처리 작업 자체가 단순함
        - 처리가 단순하여 과부화 걸릴 가능성이 낮음
      - 상위 계층의 버퍼 활용상위 계층이 자체 버퍼(임시 저장 공간)를 가지고 있음
        - 네트워크 계층이 데이터를 넘기면 일단 버퍼에 쌓아둠
        - 상위 계층이 그 순간 바쁘더라도 버퍼 덕분에 데이터가 유실되지 않음
        - 받는 즉시 처리하지 않고 여유 있을 때 꺼내어 처리 가능
      - 상위 계층 프로토콜의 자체 흐름 제어
        - 상위 계층 프로토콜(e.g., TCP) 대부분이 이미 흐름 제어 기능 소유
        - 네트워크 계층에서 흐름 제어를 또 추가하면 기능 중복
        - 시스템 복잡성만 증가하고 전체 효율성은 오히려 저하

*흐름 제어 (Flow Control)
발신지가 수신자를 과부하를 주지 않도록 데이터 전송량을 조절하는 것.
*버퍼 (Buffer)
메모리 상의 임시 저장 공간.

네트워크 계층의 소개

- 네트워크 계층 문제점
  - 종류 (3/4)
    - 혼잡 발생 가능성 존재
      - 인터넷 내부에 데이터그램이 너무 많이 존재하여 혼잡 상황이 발생

      - 비연결형 네트워크에서의 혼잡 제어 해결책
        - 후방 시그널링과 전방 시그널링 사용
        - 라우터가 혼잡을 탐지할 때 송신자에게 전송되는 ICMP를 사용하여 초크 패킷(Choke Packet) 사용
        - 메시지의 중요도에 따라 패킷에 우선순위 부여하여 혼잡 시 낮은 우선순위 패킷은 폐기

      - 연결형 네트워크에서의 혼잡 제어 해결책
        - 추가 가상회선 생성
        - 설정 과정(Setup Phase)에서 가상 회선 설정 시 트래픽 수준 합의

*후방 시그널링(Backward Signaling)
혼잡 발생 방향과 반대 방향으로 전달되는 BECN(Backward Explicit Congestion Notification) 비트를 사용하여 송신자에게 알림
*전방 시그널링(Forward Signaling)
혼잡 발생 방향과 같은 방향으로 전달되는 FECN(Forward Explicit Congestion Notification) 비트를 사용하여 수신자에게 알림.

*초크 패킷(Choke Packet)
혼잡이 발생했을 때, 발신지에 트래픽 전송 속도를 줄이도록 경고하기 위해 보내는 제어용 패킷.

네트워크 계층의 소개

- 네트워크 계층 문제점
  - 종류 (4/4)
    - 서비스 품질 (QoS, Quality of Service) 저하
      - 서비스마다 최적화된 품질 제공의 어려움
        - 대역폭 부족·지연·혼잡·오류로 인한 서비스 불안정 및 품질 저하
    - 라우팅 성능 저하
      - 네트워크 규모 확대, 경로 변경 등에 따른 성능 저하 문제
        - 규모 증가에 따른 라우팅 테이블 크기 증가, 메모리·처리 능력 부족 시 성능 저하
    - 보안 문제
      - 데이터 무결성 및 기밀성 부족
        - 암호화되지 않거나 신뢰할 수 없는 네트워크에서의 변조·도청 위험
      - 패킷 분할·재조립 취약성
        - 분할·재조립 과정에서 공격자의 패킷 변형·삭제 위험

네트워크 계층의 소개

- 네트워크 계층 보안 문제 해결책 (1/6)
  - IPsec (Internet Protocol security)
    - 정의
      - 네트워크 계층에서 IP 패킷에 인증, 무결성, 기밀성을 제공하기 위한 보안 프로토콜 집합
    - 필요성
      - IP는 기본적으로 암호화 기능이 없어 전송 데이터의 기밀성 보장이 필요
      - IP는 데이터 변조 여부 확인과 송/수신자 인증 기능이 없어 무결성과 인증 보장이 필요
    - 특징
      - 네트워크 계층에서 IP 패킷을 보호하여 보안 통신 채널 제공
      - 가상 사설망(VPN)에서 주로 사용

네트워크 계층의 소개

- 네트워크 계층 보안 문제 해결책 (2/6)
  - IPsec (Internet Protocol security)
    - 운영 모드
      - 목적
        - 통신 대상과 환경에 따라 보호할 IP 패킷 범위를 결정하기 위해 사용
      - 특징
        - IPsec 연결 설정(IKE 협상, Internet Key Exchange) 시 먼저 결정
        - 선택된 모드에 따라 SA(Security Association)가 생성
        - 설정된 SA는 통신 중 유지되며, 보호 방식 변경 시 새로운 SA 생성 필요
      - 종류
        - 전송 모드(Transport Mode)
        - 터널 모드(Tunnel Mode)

*IKE (Internet Key Exchange)
IPsec 통신에 필요한 보안 알고리즘과 키를 협상하는 프로토콜.
*SA (Security Association)
IPsec 통신에서 사용할 보안 정책과 키 정보를 저장한 논리적 연결.

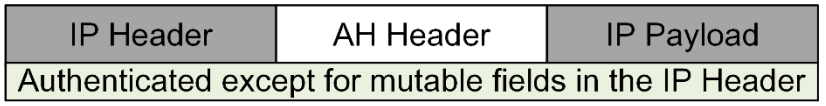
네트워크 계층의 소개

- 네트워크 계층 보안 문제 해결책 (3/6)
  - IPsec (Internet Protocol security)
    - 운영 모드
      - 전송 모드(Transport Mode)
        - 정의
          - 원본 IP 헤더는 유지하고 IP Payload 부분에 보안 기능을 적용하는 방식
        - 특징
          - 호스트-호스트 간에 주로 사용
          - IP 헤더를 그대로 사용하여 오버헤드가 적음
      - 터널 모드(Tunnel Mode)
        - 정의
          - 원래 IP 패킷 전체를 암호화 및 인증하고, 새로운 IP 헤더를 추가하여 전송하는 방식
        - 특징
          - 라우터-라우터 또는 호스트-라우터 간 VPN에 주로 사용
          - 내부 IP 주소를 보호할 수 있음

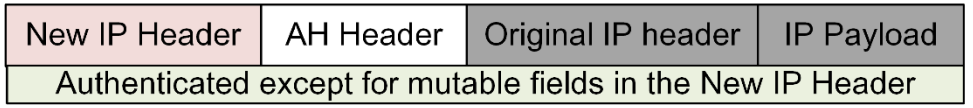
네트워크 계층의 소개

- 네트워크 계층 보안 문제 해결책 (4/6)
  - IPsec (Internet Protocol security)
    - 핵심 프로토콜 (1/2)
      - AH(Authentication Header) 프로토콜
        - 전송 모드
          - 변경 가능한 IP Header 필드를 제외한 IP Packet 전체에 대한 무결성과 인증 제공
          - 데이터 암호화 기능은 제공하지 않음
        - 터널 모드
          - 원본 IP Packet 전체와 새로운 IP Header 일부를 제외한 전체 데이터에 대한 무결성과 인증 제공
          - 데이터 암호화 기능은 제공하지 않음

AH
Transport Mode

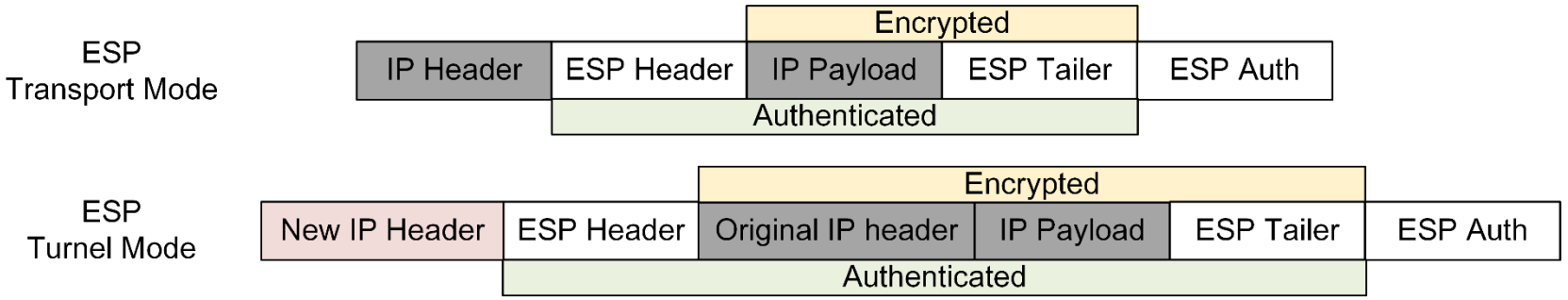


AH
Turnel Mode



네트워크 계층의 소개

- 네트워크 계층 보안 문제 해결책 (5/6)
  - IPsec (Internet Protocol security)
    - 핵심 프로토콜 (2/2)
      - ESP(Encapsulating Security Payload) 프로토콜
        - 전송 모드
          - IP Payload를 암호화하고 ESP Header와 암호화 데이터를 인증
          - 데이터 기밀성, 무결성 및 발신지 인증 제공
        - 터널 모드
          - 원본 IP Packet 전체를 암호화하고 ESP Header와 암호화 데이터를 인증
          - 내부 IP 주소 보호 및 VPN 환경에서 주로 사용



네트워크 계층의 소개

- 네트워크 계층 보안 문제 해결책 (6/6)
  - IPsec (Internet Protocol security)
    - 제공 서비스
      - 알고리즘과 키의 결정
        - IKE를 통해 암호 알고리즘과 세션 키를 협상
      - 패킷 암호화
        - 협상된 암호 알고리즘과 키를 이용해 전송 패킷을 암호화
        - 패킷 스니핑 공격 무력화
      - 데이터 무결성
        - 전송 중 패킷이 변조되지 않았음을 보장
        - 무결성 검증 실패 시 수신 패킷 폐기
        - 패킷 수정 공격 방지
      - 발신지 인증
        - 패킷이 위장자에 의해 생성되지 않았음을 확인
        - IP 스푸핑 공격 방지

목 차

- 네트워크 계층의 소개
- IPv4 주소

IPv4 주소

- IPv4 (Internet Protocol version 4) (1/7)
  - 정의
    - 네트워크에서 호스트 간의 데이터 전송을 위해 경로 및 목적지를 지정하는 32비트 체계의 고유한 주소

*주소 공간(Address Space)
프로토콜에서 사용되는 주소의 총 개수.

  - 특징 (1/2)
    - 주소 공간(Address Space)을 지님
      - IPv4는 32비트 사용으로 2^{32}(약 43억 개)의 주소 공간 보유
      - 0.0.0.0부터 255.255.255.255까지 정의
    - 유일
      - 두 장치가 같은 네트워크에서 동시에 같은 주소를 가질 수 없음
      - 한 장치가 두 네트워크에 연결되면 주소 2개 보유 가능
    - 보편적
      - 인터넷에 연결하려는 모든 호스트가 준수해야 하는 주소 체계 의미

IPv4 주소

- IPv4 (Internet Protocol version 4) (2/7)
  - 특징 (2/2)

*기수
어떤 수 표기법에서 각 자리가 나타낼 수 있는 서로 다른 숫자의 개수.

    - 3가지 표기법(Notation) 존재
      - 2진 표기법(Binary Notation) – 기수 2
        - 비트 단위 0과 1로 주소를 표현
        - e.g., 10000001.00001011.00001011.11101111
      - 점 10진 표기법(Dotted-demical Notation) – 기수 256
        - 한 바이트 단위로 10진수로 표현
        - e.g., 129.11.11.239
      - 16진 표기법(Hexadecimal Notation) – 기수 16
        - 4비트 단위로 16진수로 표현
        - e.g., 0X810B0BEF 또는 $810B0B0EF_{16}$

2진수 자릿값	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
2진 표기법(10000001_2)	1	0	0	0	0	0	0	1
계산	$2^7 \times 1$	$2^6 \times 0$	$2^5 \times 0$	$2^4 \times 0$	$2^3 \times 0$	$2^2 \times 0$	$2^1 \times 0$	$2^0 \times 1$
점 10진 표기법 (129_{10})	129							
16진 표기법(81_{16})	$129 = 8 \times 16^1 + 1 \times 16^0$							

IPv4 주소

- IPv4 (Internet Protocol version 4)
  - 예제 5.1

다음 2진 표기법 IPv4 주소를 점 10진 표기법으로 변환하라.

a. 10000001.00001011.00001011.11101111
b. 11000001.10000011.00011011.11111111
c. 11100111.11011011.10001011.01101111
d. 11111001.10011011.11111011.00001111

- 풀이
  - 10000001_2
    $= (1 \times 2^7) + (0 \times 2^6) + (0 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 129_{10}$

a. $(129.11.11.239)_{10}$
b. $(193.131.27.255)_{10}$
c. $(231.219.139.111)_{10}$
d. $(249.155.251.15)_{10}$

IPv4 주소

- IPv4 (Internet Protocol version 4)
  - 예제 5.2

다음 10진 표기법 IPv4 주소를 2진 표기법으로 변환하라.

a. 111.56.45.78
b. 221.34.7.82
c. 241.8.56.12
d. 75.45.34.78

- 풀이
  - $111_{10} = (0 \times 2^7) + (1 \times 2^6) + (1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$
    $\rightarrow 01101111_2$

a. $(01101111.00111000.00101101.01001110)_2$
b. $(11011101.00100010.00000111.01010010)_2$
c. $(11110001.00001000.00111000.00001100)_2$
d. $(01001011.00101101.00100010.01001110)_2$

IPv4 주소

- IPv4 (Internet Protocol version 4)
  - 예제 5.3

다음 IPv4 주소 표기법에서 잘못된 점을 무엇인가.
a. 111.56.045.78
b. 221.34.7.8.20
c. 75.45.301.14
d. 11100010.23.14.67

- 풀이

  a. 점 10진 표기법에서 0이 맨 앞에 나와있으면 안됨
  b. IPv4 주소에서는 4바이트보다 많으면 안됨
  c. 점 10진 표기법에서 각 숫자는 255보다 작거나 같아야 함 $(301 > 255)$
  d. 2진 표기법과 점 10진 표기법을 혼합해서 사용하는 것은 허용하지 않음

IPv4 주소

- IPv4 (Internet Protocol version 4)
  - 예제 5.4

다음 2진 표기법 IPv4 주소를 16진 표기법으로 변환하라.
a. 10000001.00001011.00001011.11101111
b. 11000001.10000011.00011011.11111111

- 풀이
  - 16진 표기법은 각 숫자 4비트에 해당
    - 0~15(10진수)까지 표기 가능 (0~F)
  - 2진 표기법에서 16진 표기법으로 변환 시, 2진 → 10진 → 16진 순으로 변환
    - 2진 → 10진 으로 바꿀 때, 8비트 단위가 아닌 4비트씩 쪼개어 계산

a. $(10000001.00001011.00001011.11101111)_2 \rightarrow (0x810B0BEF)_{16}$

b. $(11000001.10000011.00011011.11111111)_2 \rightarrow (0xC1831BFF)_{16}$

IPv4 주소

- IPv4 (Internet Protocol version 4)
  - 예제 5.5

처음 주소가 146.101.29.0이고 마지막 주소가 146.102.32.255인 경우 이 범위 내의 주소의 수를 찾아보라

- 풀이
  - 마지막 주소 – 처음 주소 = 146.102.32.255 – 146.101.29.0 = 0.1.3.255
  - 주소의 개수 $= (0 \times 256^3 + 0 \times 256^2 + 3 \times 256^1 + 255 \times 256^0) + 1 = 1{,}024$
    - 1은 첫 주소와 마지막 주소를 포함하기 때문에 더하기 계산

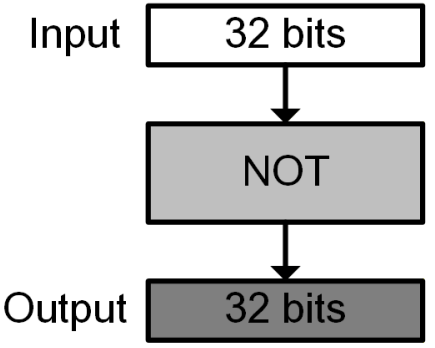
  - 예제 5.6

주소 범위의 첫 주소는 14.11.45.96이다. 만약, 범위 내 주소의 수가 32라면 마지막 주소는 무엇인가?

- 풀이
  - 주소의 수에서 1을 뺀 후 그 결과를 기수 256의 수로 변환하면 0.0.0.31
  - 이 값을 첫 주소에 더하여 마지막 주소 계산
    - 덧셈은 기수 256으로 수행
  - 마지막 주소 = $(14.11.45.96 + 0.0.0.31)_{256}$ = 14.11.45.127

IPv4 주소

- IPv4 (Internet Protocol version 4) (3/7)
  - 연산(Operations)의 종류 (1/3)
    - NOT 연산
      - 2진 표기법일 경우
        - 0 ↔ 1
      - 점 10진 표기법일 경우
        - 255에서 바이트 값을 뺀 값
      - 예제 5.7



NOT

Input	Output
0	1
1	0

Operation for each bit

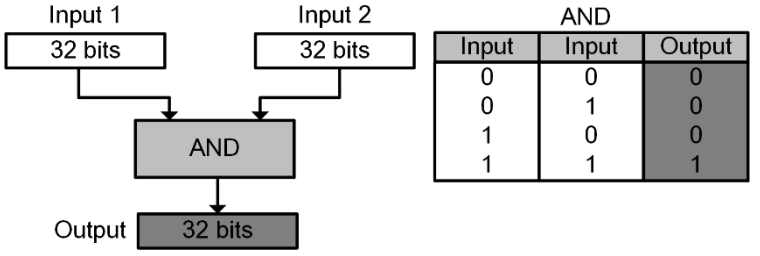
다음 이진법 혹은 점 10진 표기법으로 표기된 32비트 수에 NOT 연산을 적용하여라.

a. 00010001 01111001 00001110 00100011
b. 17.121.14.35

- 풀이
  a. 11101110.10000110.11110001.11011100
  b. 238.134.241.220

IPv4 주소

- IPv4 (Internet Protocol version 4) (4/7)
  - 연산(Operations)의 종류 (2/3)
    - AND 연산
      - 2진 표기법일 경우
        - 둘 다 1일 경우에만 1
      - 점 10진 표기법일 경우
        - 둘 중 하나가 0 또는 255일 경우 작은 값
        - 두 바이트가 0 또는 255가 아닐 경우
          - 각 바이트를 2진수 자릿값으로 계산하여 작은 항을 취하고 모두 더한 값
      - 예제 5.8



AND		
Input	Input	Output
0	0	0
0	1	0
1	0	0
1	1	1

다음 이진법(a, b) 혹은 점 10진 표기법(c, d)으로 표기된 두 개의 32비트 수에 AND 연산을 적용하여라.

a. 00010001 01111001 00001110 00100011
b. 11111111 11111111 10001100 00000000
c. 17.121.14.35
d. 255.255.140.0

- a, b 풀이

a	00010001.01111001.00001110.00100011
b	11111111.11111111.10001100.00000000
결과	00010001.01111001.00001100.00000000

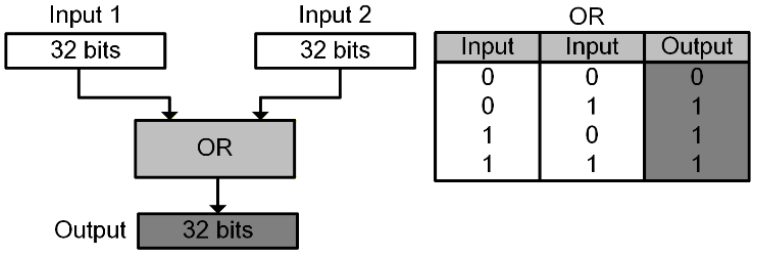
- c, d 풀이

c	17.121.14.35
d	255.255.140.0
결과	17.121.12.0

c(14)	0	0	0	0	8	4	0	0
d(140)	128	0	0	0	8	4	0	0
결과(12)	0	0	0	0	8	4	0	0

IPv4 주소

- IPv4 (Internet Protocol version 4) (5/7)
  - 연산(Operations)의 종류 (3/3)
    - OR 연산
      - 2진 표기법일 경우
        - 둘 중 하나라도 1일 경우에 1
      - 점 10진 표기법일 경우
        - 둘 중 하나가 0 또는 255일 경우 큰 값
        - 두 바이트가 0 또는 255가 아닐 경우
          - 각 바이트를 2진수 자릿값으로 계산하여 큰 항을 취하고 모두 더한 값
      - 예제 5.9



OR		
Input	Input	Output
0	0	0
0	1	1
1	0	1
1	1	1

다음 이진법(a, b) 혹은 점 10진 표기법(c, d)으로 표기된 두 개의 32비트 수에 OR 연산을 적용하여라.

a. 00010001 01111001 00001110 00100011
b. 11111111 11111111 10001100 00000000
c. 17.121.14.35
d. 255.255.140.0

- a, b 풀이

a	00010001.01111001.00001110.00100011
b	11111111.11111111.10001100.00000000
결과	11111111.11111111.10001110.00100011

- c, d 풀이

c	17.121.14.35
d	255.255.140.0
결과	255.255.142.35

c(14)	0	0	0	0	8	4	2	0
d(140)	128	0	0	0	8	4	0	0
결과(142)	128	0	0	0	8	4	2	0

IPv4 주소

- IPv4 (Internet Protocol version 4) (6/7)
  - 구조
    - 네트워크 ID (Network ID)
      - 정의
        - 주소가 어떤 네트워크에 속해 있는지를 식별하는 주소 파트
      - 특징
        - 데이터를 목적지로 보내기 위해 네트워크 대역을 찾는 데 사용
    - 호스트 ID (Host ID)
      - 정의
        - 네트워크 안에서 개별 기기를 식별하기 위한 고유 주소 파트
      - 특징
        - 네트워크 ID가 같은 기기라도 호스트 ID는 서로 달라야 충돌 없이 통신 가능

IPv4 주소

- IPv4 (Internet Protocol version 4) (7/7)
  - 주소 지정
    - 정의
      - 인터넷 상 단말기들을 식별하기 위해 32비트 주소를 부여하는 방식
    - 특징
      - 주소를 네트워크 ID와 호스트 ID의 두 부분으로 나누어 관리
    - 종류
      - 클래스기반 주소 지정
        - 주소 관리를 용이하게 하기 위해 크기 별로 클래스를 나누어 주소 할당
      - 클래스없는 주소 지정
        - 주소 낭비를 줄이고자 클래스 단위를 없애고 비트 단위로 주소를 쪼개어 할당

*클래스(Class)
IP 주소를 효율적으로 분배하기 위해 미리 정해둔 사이즈별 고정 규격.

IPv4 주소

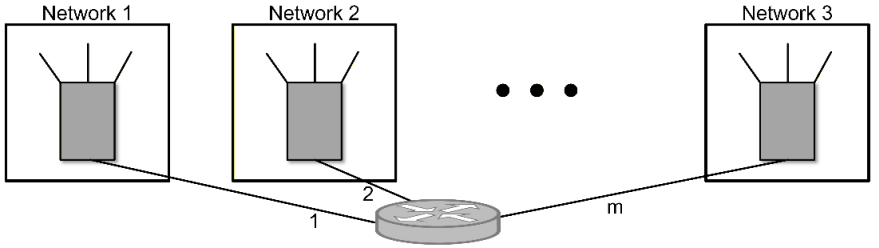
- 클래스기반 주소지정
  - 정의
    - IP 주소 공간을 5개의 클래스(A, B, C, D, E)으로 나누어 배분하는 방법
  - 특징
    - 2계층 주소 지정을 사용한 유연한 네트워크 크기 분배
      - 클래스 A, B, C의 네트워크 ID를 8비트(대규모), 16비트(중규모), 24비트(소규모)로 다양화
    - 네트워크 그룹 생성 가능
      - 네트워크 개수가 폭발적으로 증가되는 상황에서 기존 256개 제한에서 벗어나 더 많은 네트워크 그룹 생성 가능
    - 식별 자동화
      - 클래스를 통해 주소가 속한 네트워크 크기를 라우터가 파악하기 쉬워 주소 탐색 속도 증가

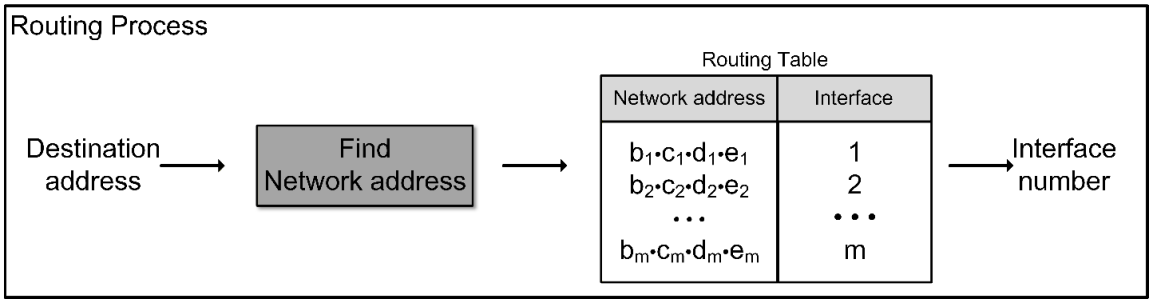
IPv4 주소

- 클래스기반 주소지정
  - 주요 개념 (1/3)
    - 클래스(Class)
      - 정의
        - IP 주소를 효율적으로 분배하기 위해 미리 정해둔 사이즈별 고정 규격
      - 특징
        - 주소 성격과 규모에 따라 총 5개의 등급(A, B, C, D, E)으로 대분류
    - 블록(Block)
      - 정의
        - 동일한 네트워크 식별자(Net ID)를 공유하는 연속된 IP 주소들의 집합
      - 특징
        - 클래스 소속(A, B, C)에 따라 주소 공간의 크기가 기계적으로 고정

IPv4 주소

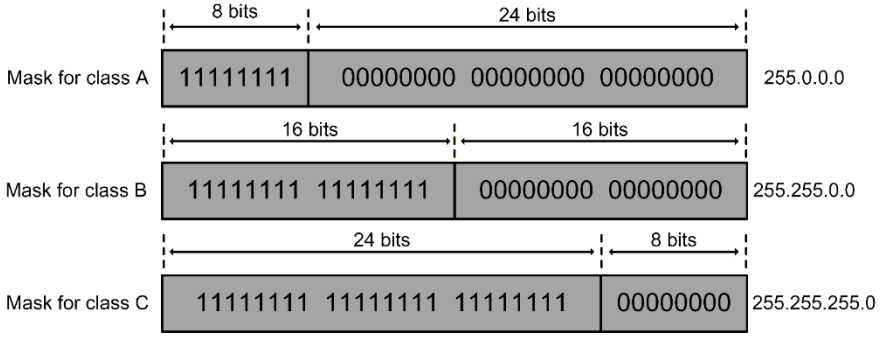
- 클래스기반 주소지정
  - 주요 개념 (2/3)
    - 네트워크 주소
      - 정의
        - 주소가 어떤 네트워크 블록에 속해있는 지를 식별하는 주소 파트
      - 특징
        - 호스트 ID 파트가 전부 0인 주소
        - 라우터는 목적지 주소를 통해 네트워크 주소를 찾아 인터페이스 번호 탐색

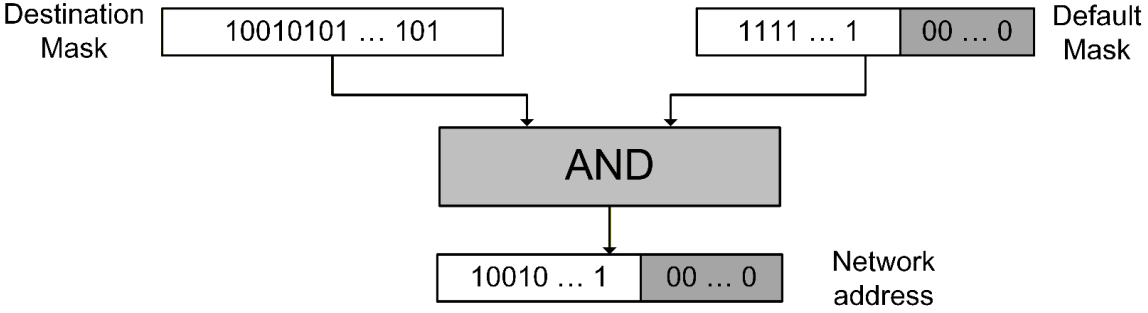




IPv4 주소

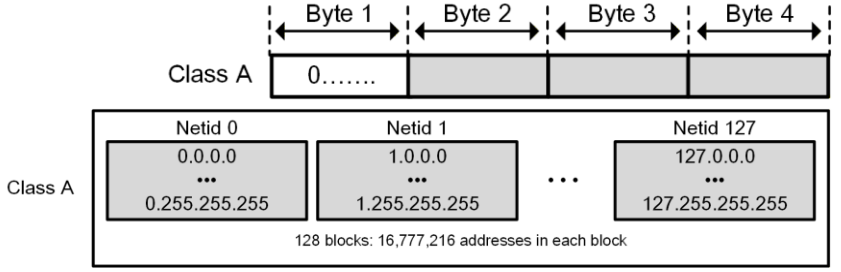
- 클래스기반 주소지정
  - 주요 개념 (3/3)
    - 네트워크 마스크
      - 정의
        - 네트워크 ID와 호스트 ID를 비트 단위로 구분 비트 필터
      - 특징
        - 네트워크 ID가 모두 1이고 호스트 ID가 모두 0인 주소
        - 네트워크 마스크와 AND 연산을 통해 네트워크 주소 혹은 목적지 주소를 계산

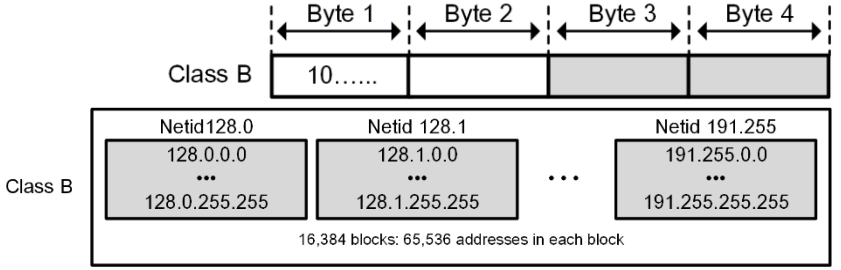


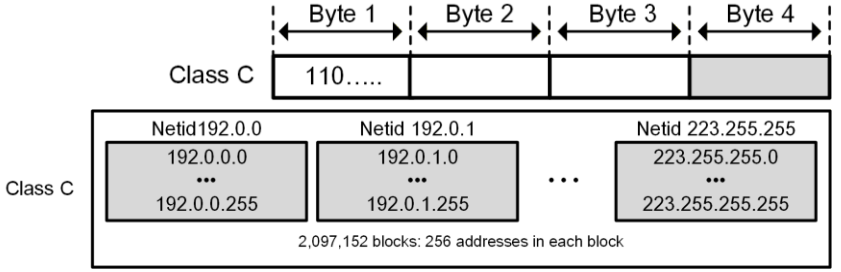


IPv4 주소

- 클래스기반 주소 지정
  - 클래스 종류 (1/2)
    - 클래스 A
      - 초대형 기관에 할당하기 위해 설계
        - e.g., 정부 부처
      - 비트 식별자 0, Netid = 8bits
    - 클래스 B
      - 중대형 기관에 할당하기 위해 설계
        - e.g., 대기업, 대학교, 대형 ISP
      - 비트 식별자 10, Netid = 16bits
    - 클래스 C
      - 소규모 기관에 할당하기 위해 설계
        - e.g., 중소 기업, 연구소, 일반 가정집
      - 비트 식별자 110, Netid = 24bits





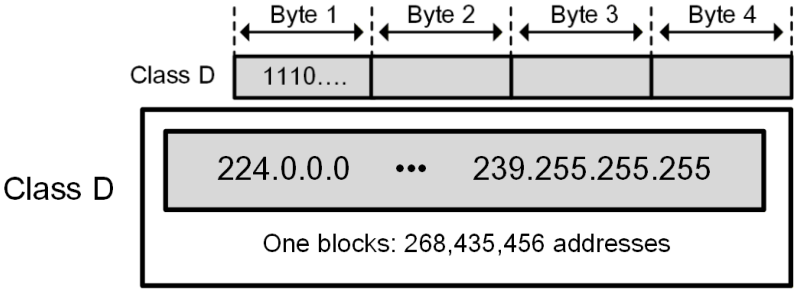


IPv4 주소

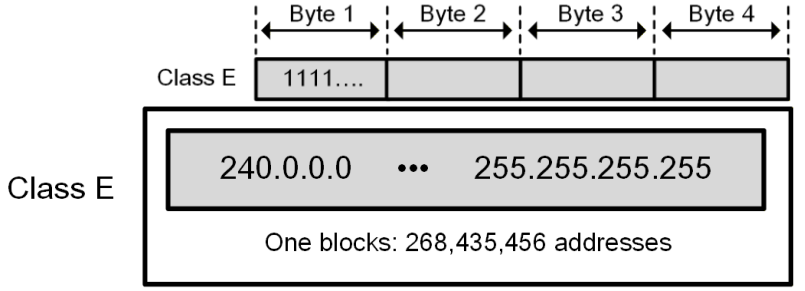
- 클래스기반 주소 지정

*멀티캐스팅(Multicasting)
하나의 주소로 그룹에 속한 모든 호스트에게 전달되는 통신 방식.

  - 클래스 종류 (2/2)
    - 클래스 D
      - 일대다 통신을 위한 멀티캐스트(Multicast) 전용 주소
        - e.g., 화상 회의, 인터넷 방송
      - 비트 식별자 1110
      - 네트워크 ID와 호스트 ID가 아닌 하나의 블록으로 존재

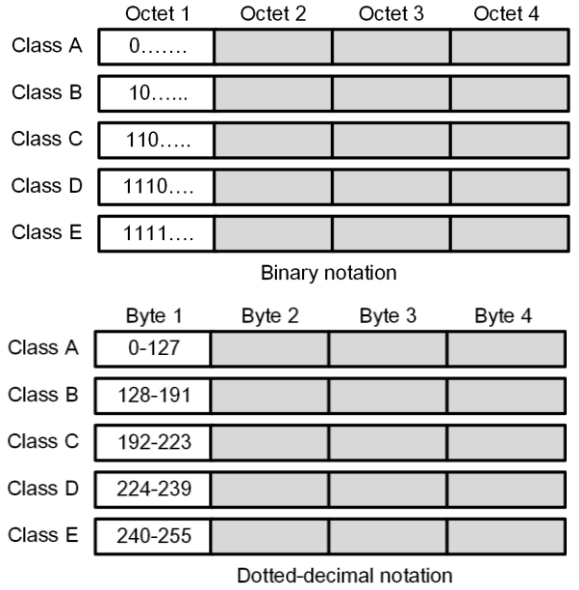


    - 클래스 E
      - 예약된 주소로 사용하기 위해 설계
        - e.g., 255.255.255.255
      - 비트 식별자 1111
      - 네트워크 ID와 호스트 ID가 아닌 하나의 블록으로 존재



IPv4 주소

- 클래스기반 주소지정
  - 클래스 탐색
    - 주소가 2진 표기법일 경우
      - 맨 왼쪽의 몇 개 비트로 판단
    - 주소가 점 10진 표기법일 경우
      - 첫 번째 바이트 범위 확인
    - 예제 5.10



각 주소의 클래스를 나타내어라.

A.	00000001 00001011 00001011 11101111	a.	227.12.14.87
B.	11000001 10000011 00011011 11111111	b.	193.14.56.22
C.	10100111 11011011 10001011 01101111	c.	14.23.120.8
D.	11110011 10011011 11111011 00001111	d.	252.5.15.111

- 풀이

A	B	C	D	a	b	c	d
클래스 A	클래스 C	클래스 B	클래스 E	클래스 D	클래스 C	클래스 A	클래스 E

IPv4 주소

- 클래스기반 주소지정
  - 예제 5.11

각각의 블록의 주소를 통해, 블록 내의 주소 수, 첫 주소(네트워크 주소)와 마지막 주소를 구하라.

1. 73.22.17.25
2. 180.8.17.9
3. 200.11.8.45

- 풀이

1. 73.22.17.25
- 73은 0과 127 사이의 값이므로 클래스 A이므로 n값은 8
- 주소 수 $N = 2^{32-n} = 2^{24} = 16{,}777{,}216$
- 왼쪽의 8비트를 유지, 오른쪽 24비트를 0으로 구성
  - 첫 주소(네트워크 주소)는 73.0.0.0
- 왼쪽의 8비트를 유지하고 오른쪽 24비트를 1로 구성
  - 마지막 주소는 73.255.255.255

2. 180.8.17.9
- 180은 128과 192 사이의 값이므로 클래스 B이므로 n값은 16
- 주소 수 $N = 2^{32-n} = 2^{16} = 65{,}536$
- 왼쪽의 16비트를 유지, 오른쪽 16비트를 0으로 구성
  - 첫 주소(네트워크 주소)는 180.8.0.0
- 왼쪽의 8비트를 유지, 오른쪽 24비트를 1로 구성
  - 마지막 주소는 180.8.255.255

3. 200.11.8.45
- 200은 192과 223사이의 값이므로 클래스 C이므로 n값은 24
- 주소 수 $N = 2^{32-n} = 2^{8} = 256$
- 왼쪽의 24비트를 유지, 오른쪽 8비트를 0으로 구성
  - 첫 주소(네트워크 주소)는 200.11.8.0
- 왼쪽의 24비트를 유지하고 오른쪽 8비트를 1로 구성
  - 마지막 주소는 200.11.8.255

IPv4 주소

- 클래스기반 주소지정
  - 예제 5.12

라우터가 목적지 주소가 201.24.67.32인 패킷을 받는다. 이 주소의 클래스가 B 일 때, 라우터가 패킷의 네트워크 주소를 찾는 방법을 보여라.

- 풀이
  - 목적지 주소에서 네트워크 주소를 구할 때, 네트워크 마스크와 AND 연산
  - 클래스 B의 네트워크 마스크는 255.255.0.0
    - 0또는 255일 경우 작은 값 선택

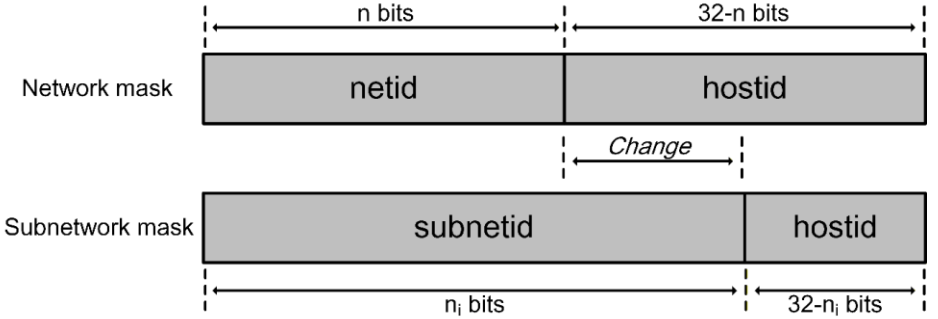
목적지 주소	201.24.67.32
디폴트 마스크	255.255.0.0
네트워크 주소	201.24.0.0

IPv4 주소

- 클래스기반 주소지정
  - 3계층 주소 지정: 서브네팅(Subnetting) (1/2)
    - 개요
      - 클래스 A나 B 블록을 받은 조직의 보안과 관리 강화
        - 세부적으로 관리하기 위해 서브네트워크 분할 필요
      - 클래스 A나 B 블록은 대부분 소진되었으며 클래스 C 블록은 조직이 필요로 하는 크기보다 작은 크기
        - 서브블록 분할 후 타 조직과 공유 가능

IPv4 주소

- 클래스기반 주소지정
  - 3계층 주소 지정: 서브네팅(Subnetting) (2/2)
    - 정의
      - 블록을 작은 블록으로 나누는 개념
    - 특징
      - 서브넷 마스크
        - 네트워크를 몇 개의 서브 네트워크로 나눌 경우 생성
        - $n_{sub} = n + \log_2 S$
          (n_{sub}는 subnetid, n은 netid 길이, S는 서브넷의 개수이며 2의 거듭제곱)



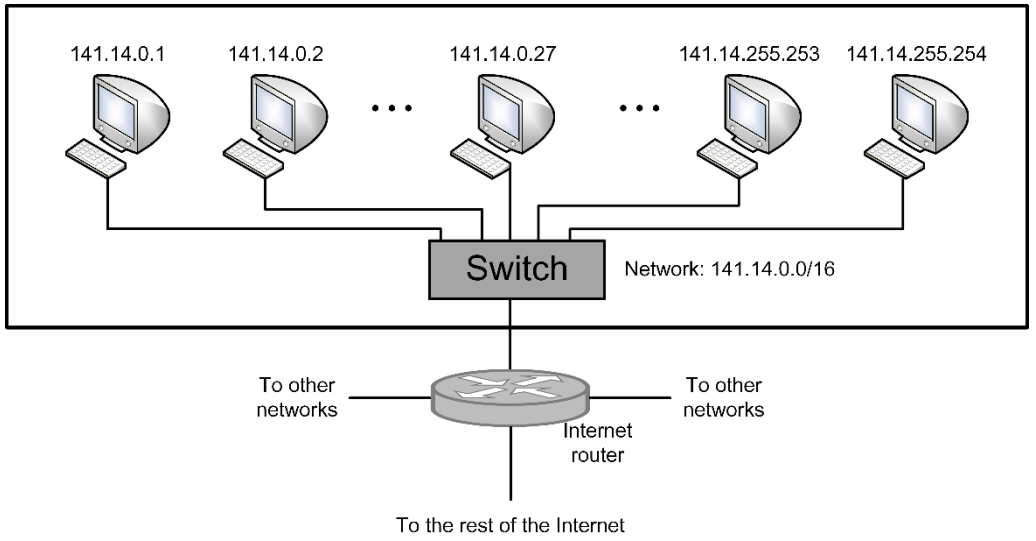
      - 서브넷 주소
        - 서브넷의 첫 주소는 서브넷의 식별자 역할
          - 해당하는 서브 네트워크로 가는 패킷을 전달하기 위해 사용
        - 서브 네트워크 주소를 찾을 때 주어진 주소와 서브넷 주소 AND 연산
          - 네트워크 주소 추출 시 사용한 AND 연산과 동일

IPv4 주소

- 클래스기반 주소지정
  - 예제 5.13

그림에 대해 설명하라.

- 풀이
  - 클래스 B netid의 길이를 보이기 위하여 /16을 사용
  - /16을 통해 클래스 B 주소를 사용하는 네트워크가 서브네팅을 사용하기 전의 구성
  - 거의 2^{16}개의 호스트를 가진 한 개의 네트워크 만을 사용
  - 한 개의 연결을 통하여 네트워크가 인터넷 내의 한 개의 라우터에 연결

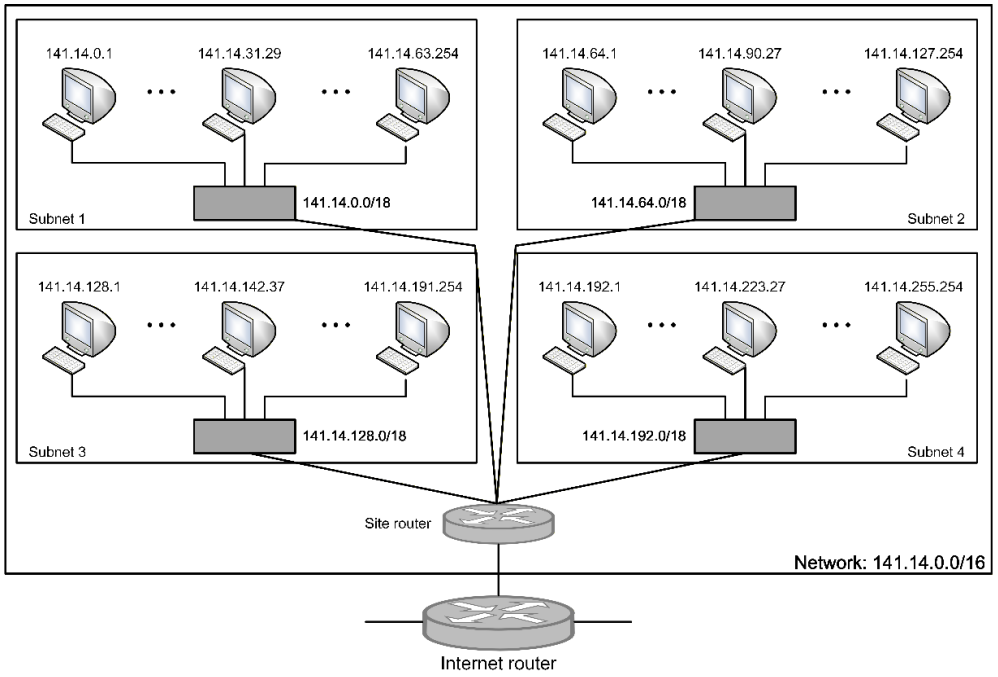


IPv4 주소

- 클래스기반 주소지정
  - 예제 5.14

그림을 보고 서브넷 마스크와 클래스 B의 마스크인 255.255.0.0이 다름을 설명하고,
서브넷 2의 주소 중 하나가 141.14.120.77일 때, 서브넷 주소를 찾아보아라.

- 풀이
  - 141.14.0.0 ~ 141.14.255.255 대역의 네트워크를 네 개의 서브넷으로 나뉘면
    $(\text{네트워크 마스크})n = 16, \quad (\text{서브넷 마스크})n_1 = n_2 = n_3 = n_4 = 16 + \log_2 4 = 18$
  - 서브넷 마스크가 18개의 1을 가지고
    14개의 0을 가짐을 의미함
    - 255.255.192.0
  - 클래스 B의 마스크
    - 255.255.0.0



  - 서브넷 2의 서브넷 주소

	주소	141.14.120.77						
	마스크	255.255.192.0						
	서브넷 주소	141.14.64.0						
주소(120)	0	64	32	16	8	0	0	0
마스크(192)	128	64	0	0	0	0	0	0
결과(64)	0	64	0	0	0	0	0	0

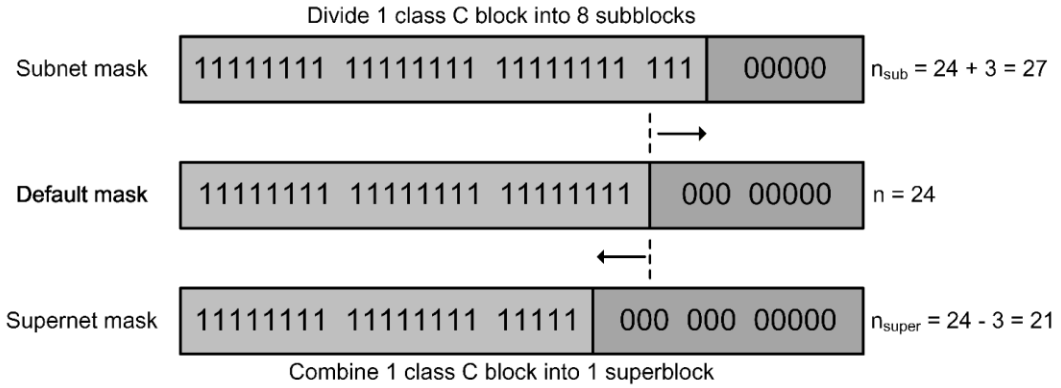
IPv4 주소

- 클래스기반 주소지정
  - 슈퍼네팅(Supernetting) (1/3)
    - 개요
      - 3계층 주소 지정의 한계
        - 대부분의 조직이 블록 공유를 원하지 않아 서브네팅으로도 해결 안 되는 주소 고갈 문제
        - 클래스 C 블록은 남아 있으나 조직 요구사항 충족에는 부족한 크기

IPv4 주소

- 클래스기반 주소지정
  - 슈퍼네팅(Supernetting) (2/3)
    - 정의
      - 여러 개의 작은 네트워크를 하나의 큰 네트워크로 통합하는 기술
    - 슈퍼넷 마스크
      - 정의
        - 슈퍼네팅(Supernetting)에 사용되는 마스크
      - 계산
        - n_{super}는 supernetid의 길이, c는 결합된 클래스 C 블록의 개수

$$n_{super} = n - log_2 c$$



IPv4 주소

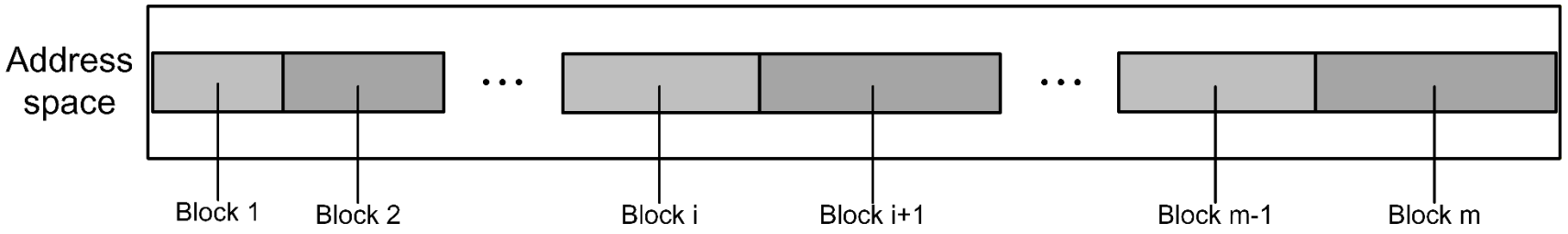
- 클래스기반 주소지정
  - 슈퍼네팅(Supernetting) (2/3)
    - 효과
      - 여러 개의 작은 네트워크를 통합하여 IP 주소를 효율적으로 관리
      - 라우팅 테이블 정보량 감소로 컴퓨팅 자원 절약
    - 문제점
      - 2의 거듭제곱 블록 단위 할당으로 인한 주소 낭비
        - e.g., 7개의 블록을 필요로 하는 조직은 8개의 블록 할당
      - 패킷 라우팅의 복잡화

IPv4 주소

- 클래스기반 주소지정
  - 문제점
    - 주소 고갈 문제
      - 클래스의 고정된 크기 때문에 작은 네트워크가 클래스 A를 사용하면 많은 IP 주소가 낭비
  - 간접적인 해결책
    - 서브네팅과 슈퍼네팅
      - 주소의 배분이나 경로 설정 과정이 훨씬 복잡
  - 장기적인 해결책
    - IPv6 (Internet Protocol version 6)
  - 단기적인 해결책
    - 클래스없는 주소지정

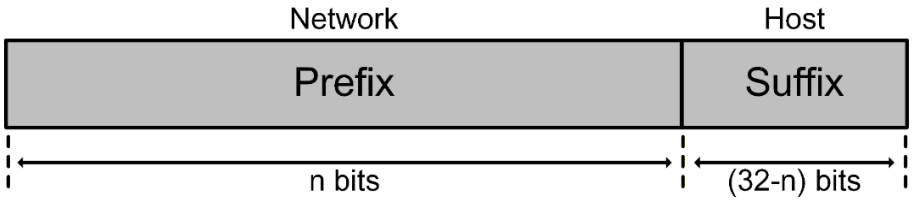
IPv4 주소

- 클래스없는 주소지정
  - 정의
    - 기존 클래스 단위 주소 지정 방법에서 클래스를 제외한 주소 지정 방법
  - 특징 (1/6)
    - 가변 길이 등록
      - 전체의 주소 공간을 가변 길이 블록으로 분할
      - 이론적으로 $2^0 \sim 2^{32}$개의 주소를 갖는 블록이 기관에게 할당 가능



IPv4 주소

- 클래스없는 주소지정
  - 특징 (2/6)
    - 프리픽스(Prefix)와 서픽스(Suffix)로 구분
      - 프리픽스(Prefix)는 netid와, 서픽스(Suffix)는 hostid와 동일한 임무 수행
      - 프리픽스의 길이로 블록 내에 있는 주소의 수 계산 가능
      - 블록 내에 있는 주소의 수는 프리픽스의 길이는 역(Inverse)의 관계
        - 프리픽스가 크다는 것은 블록의 크기가 작는 것



    - 예제 5.15

클래스없는 주소지정 방식에서는 주소 정보만을 이용하여 주소가 속하는 블록을 알 수 없다. 예를 들어 230.8.24.56은 여러 개의 블록에 속할 수 있다. 이 주소가 포함될 일부의 블록과 이 블록의 프리픽스 길이를 보여라.

프리픽스 길이	블록	프리픽스 길이	블록	프리픽스 길이	블록
16	203.8.0.0 ~ 230.8.255.255	26	203.8.24.0 ~ 230.8.24.63	29	203.8.24.56 ~ 230.8.24.63
20	203.8.16.0 ~ 230.8.31.255	27	203.8.24.32 ~ 230.8.24.63	31	203.8.24.56 ~ 230.8.24.57

IPv4 주소

- 클래스없는 주소지정
  - 예제 5.16

전체 인터넷을 4,294,967,296개의 주소를 갖는 하나의 단일 블록이라고 하면, 인터넷의 프리픽스 길이와 서픽스 길이는 얼마인가?

- 풀이
  - 프리픽스의 길이는 0, 서픽스의 길이는 32
  - 32개의 모든 비트가 단일 블록에서 $2^{32} = 4{,}294{,}967{,}296$개의 호스트를 구분하기 위해 사용

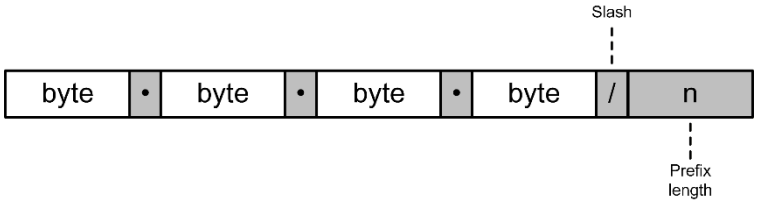
  - 예제 5.17

인터넷이 4,294,967,296개의 블록으로 나누어지고 각각의 블록은 단지 하나의 주소만을 포함하는 경우에 프리픽스 길이와 서픽스 길이는 얼마인가?

- 풀이
  - 프리픽스의 길이는 32
  - 서픽스의 길이는 0
  - $2^{32} = 4{,}294{,}967{,}296$개의 블록을 나타내기 위해 32개의 비트 모두 필요
  - 단일 주소는 블록 자체를 정의

IPv4 주소

- 클래스없는 주소지정
  - 특징 (3/6)
    - 슬래쉬(/) 표기법
      - 주소의 블록을 알기 위해서 프리픽스(Prefix)의 길이(n) 정보가 주소 정보에 포함



      - 예제 5.18

슬래쉬 표기법으로 표시된 주소들이다. 각각 주소에 대한 프리픽스 길이와 서픽스 길이를 구하라.

a. 12.23.24.78/8, 네트워크 마스크 255.0.0.0
b. 130.11.232.156/16, 네트워크 마스크 255.255.0.0
c. 167.199.170.82/27, 네트워크 마스크 255.255.255.224

- 풀이

구분	주소	네트워크 마스크	프리픽스	서픽스
a	12.23.24.78/8	255.0.0.0	8	24
b	130.11.232.156/16	255.255.0.0	16	16
c	167.199.170.82/27	255.255.255.224	27	5

IPv4 주소

- 클래스없는 주소지정
  - 블록 정보 추출 (4/6)
    - 블록에 속하는 주소의 개수

$$N = 2^{32-n}$$

    - 블록에 속하는 첫 번째 주소(네트워크 주소)
      - 블록에 속하는 임의의 주소에 네트워크 주소를 비트단위로 AND 연산으로 계산

첫 번째 주소 = (임의의 주소) AND (네트워크 마스크)

    - 블록에 속하는 마지막 주소
      - 첫 번째 주소와 주소의 개수를 더하기
      - 블록에 속하는 임의의 주소에 네트워크 마스크의 보수(NOT 연산) 값을 비트 단위로 OR 연산하여 계산

마지막 주소 = (임의의 주소) OR [NOT (네트워크 마스크)]

IPv4 주소

- 클래스없는 주소지정
  - 예제 5.19

167.199.170.82/27은 블록에 속하는 하나의 주소이다. 이 네트워크에 속하는 주소의 개수와 첫 번째 주소, 그리고 마지막 주소를 구하라.

- 풀이
  - 프리픽스 길이 n의 값은 27으로, 네트워크 마스크는 27개의 1과 5개의 0으로 구성
    - 255.255.255.240
    - 네트워크에 속하는 주소의 개수는 $2^{32-n} = 2^{32-27} = 2^5 = 32$
  - 첫 번째 주소(네트워크 주소)를 찾기 위하여 AND 계산
    - 167.199.170.64/27

이진 형태의 주소	10100111 11000111 10101010 01010010
네트워크 마스크	11111111 11111111 11111111 11100000
첫 번째 주소(AND)	10100111 11000111 10101010 01000000

  - 마지막 주소를 찾기 위해서는 네트워크 마스크에 보수를 취한 후, 그 값과 주소를 OR 계산
    - 167.199.170.95/27

이진 형태의 주소	10100111 11000111 10101010 01010010
네트워크 마스크	00000000 00000000 00000000 00011111
마지막 주소	10100111 11000111 10101010 01011111

IPv4 주소

- 클래스없는 주소지정
  - 예제 5.20

17.63.110.114/24은 블록에 속하는 하나의 주소이다. 이 네트워크에 속하는 주소의 개수와 첫 번째 주소, 그리고 마지막 주소를 구하라.

- 풀이
  - 프리픽스 길이 n의 값은 24으로, 네트워크 마스크는 24개의 1과 8개의 0으로 구성
    - 255.255.255.0
    - 네트워크에 속하는 주소의 개수는 $2^{32-n} = 2^{32-24} = 2^8 = 128$
  - 첫 번째 주소(네트워크 주소)를 찾기 위하여 왼쪽 24비트는 그대로 두고, 8비트는 0으로 구성
    - 17.63.110.0/24

이진 형태의 주소	00010001 00111111 01101110 01110010
마지막 주소	00010001 00111111 01101110 00000000

  - 마지막 주소를 찾기 위하여 왼쪽 24비트는 그대로 두고, 뒤 8비트는 모두 1로 구성
    - 17.63.110.255/24

이진 형태의 주소	00010001 00111111 01101110 01110010
마지막 주소	00010001 00111111 01101110 11111111

IPv4 주소

- 클래스없는 주소지정
  - 예제 5.21

110.23.120.14/20은 블록에 속하는 하나의 주소이다. 이 네트워크에 속하는 주소의 개수와 첫 번째 주소, 그리고 마지막 주소를 구하라.

- 풀이
  - 프리픽스 길이 n의 값은 20으로, 네트워크 마스크는 20개의 1과 12개의 0으로 구성
    - 255.255.240.0
    - 네트워크에 속하는 주소의 개수는 $2^{32-n} = 2^{32-20} = 2^{12} = 4{,}096$
  - 첫 번째 주소(네트워크 주소)를 찾기 위하여 AND 계산
    - 1110.23.112.0/24

이진 형태의 주소	01101110 00010111 01111000 00001110
네트워크 마스크	11111111 11111111 11110000 00000000
첫 번째 주소(AND)	01101110 00010111 01110000 00000000

  - 마지막 주소를 찾기 위하여 왼쪽 20비트는 그대로 두고, 뒤 12비트는 모두 1로 구성
    - 17.63.110.255/24

이진 형태의 주소	01101110 00010111 01111000 00001110
마지막 주소	01101110 00010111 01111111 11111111

IPv4 주소

- 클래스없는 주소지정

*ISP (Internet Service Provider)
개인이나 기업이 인터넷에 접속할 수 있도록 네트워크 인프라를 제공하고 관리하는 통신 사업자.

  - 특징 (5/6)
    - 블록 할당
      - ICANN (Internet Corporation for Assigned Names and Address) 라는 국제 관리 기구에서 수행
        - 개인이 아닌 ISP(Internet Service Provider) 등 대규모 기관 대상 할당
        - 3가지 제약조건
          - (조건 1) 요구 주소의 수인 N은 2의 거듭제곱
            - 프리픽스 길이인 n이 정수가 되기 위하여 필요
            - $N = 232\_^{n}$, $n = log_2(2^{32}/N) = 32 - log_2 N$
          - (조건 2) 블록에 속하는 주소의 개수로부터 프리픽스의 길이 산출 가능성
          - (조건 3) 블록은 연속적이고 미할당 주소로 구성되며 시작 주소는 요구 주소인 N의 배수

IPv4 주소

- 클래스없는 주소지정
  - 예제 5.22

ISP는 1,000개의 주소를 갖는 블록을 요청하였다. 첫 번째 주소를 18.14.12.0라고 할 때, 할당된 블록과 마지막 주소를 구하여라.

- 풀이
  - 1,000은 2의 거듭제곱이 아니므로 1,024($1{,}024 = 2^{10}$)개의 주소 할당
  - 블록의 프리픽스 길이 $n = 32 - \log_2 1024 = 32 - 10 = 22$
  - 할당된 블록은 18.14.12.0/22
  - 첫 번째 주소는 18.14.12.0/22
  - 마지막 주소는 18.14.12.255/22

  - 예제 5.23

73.0.0.0 의 클래스 A 블록을 할당받은 기관은 클래스 주소 지정에서 73.0.0.0/8 블록을 할당 받은 것으로 간주할 때, 네트워크 부분과 호스트 부분의 비트를 구하여라.

- 풀이
  - 73_{10} -> 01100001_2
  - 73.0.0.0 -> 01100001 . 00000000 . 00000000 . 00000000
  - 클래스 A는 첫 번째 바이트(8비트)가 네트워크 부분, 나머지 24비트가 호스트 부분

IPv4 주소

- 클래스없는 주소지정
  - 특징 (6/6)
    - 서브네팅(Subnetting)
      - 정의
        - 하나의 블록을 여러 개의 부-블록으로 나눈 후 각각의 부-블록을 서브네트워크에 할당
      - 서브넷 설계
        - 기관이 힐당받은 주소 N, 프리픽스 길이 n, 각 서브 네트워크에게 할당된 주소 N_{sub}, 각 서브네트워크 프리픽스 길이 n_{sub}, 서브네트워크 총 개수 s
        - 서브네트워크가 올바르게 동작하기 위한 단계

        1. 각 서브네트워크에 속하는 주소의 개수는 2의 거듭제곱
        2. 서브네트워크를 위한 프리픽스 공식 $n_{sub} = n + \log_2(N/N_{sub})$
        3. 각 서브네트워크 시작 주소는 해당 서브네트워크 주소 개수의 배수

IPv4 주소

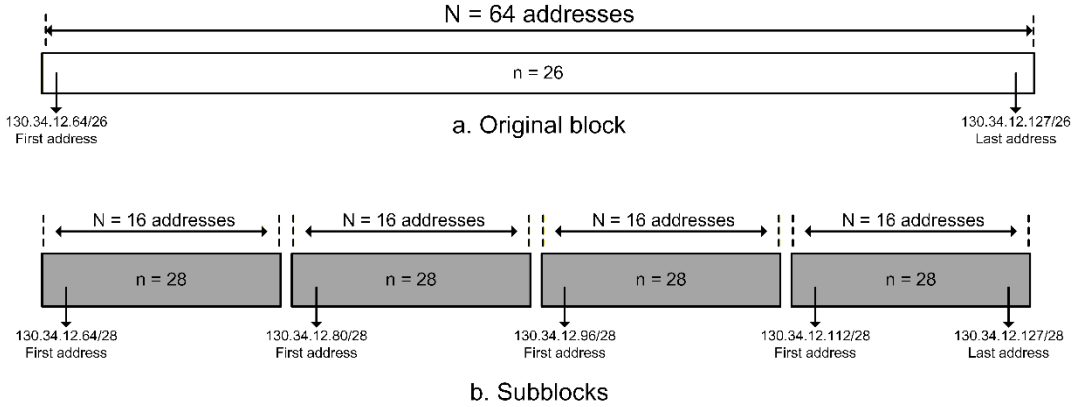
- 클래스없는 주소지정
  - 예제 5.24

130.34.12.64/26 블록이 기관에게 할당되었다. 기관은 각각이 동일한 개수의 호스트를 갖는 4개의 서브 네트워크를 구성하고자 한다. 서브 네트워크를 설계하고 각각의 서브 네트워크에 대한 정보를 구하라.

- 풀이
  - 전체 네트워크에 할당된 주소의 개수 $N = 2^{32-26} = 2^6 = 64$
  - 네트워크의 첫 번째 주소는 130.34.12.64/26
  - 네트워크의 마지막 주소는 130.34.12.127/26
  - 서브 네트워크를 위한 서브 네트워크 프리픽스 길이는

$$n_{sub} = n_1 = n_2 = n_3 = n_4 = n + \log_2(N/N_{sub}) = 26 + 2 = 28$$

  - 총 4개의 서브네트워크로 나뉘므로 한 서브 네트워크당 $N_{sub} = 2^4$개의 주소를 할당함(조건 2)
  - 각 서브 네트워크의 시작 주소는
    - 130.34.12.64/28 130.34.12.80/28
    - 130.34.12.96/28 130.34.12.112/28



IPv4 주소

- 클래스없는 주소지정
  - 예제 5.25 (1/2)

14.24.74.0/24의 시작 주소를 갖는 하나의 주소 블록이 기관에게 할당되었다. 기관은 다음과 같은 개수의 주소를 갖는 3개의 블록을 구성하고자 한다.

1. 120개의 주소를 갖는 하나의 부-블록
2. 60개의 주소를 갖는 하나의 부-블록
3. 10개의 주소를 갖는 하나의 부-블록

- 풀이
  - 이 블록에는 2^{32-24} = 256 개의 주소가 존재, 첫 번째 주소는 14.24.74.0/24, 마지막 주소는 14.24.74.255/24
  - 첫 번째 부-블록에 속하는 주소의 개수는 2의 거듭제곱이 아니기 때문에 128개 주소 할당
    - 첫 번째 주소는 네트워크 주소, 마지막 주소는 특수 목적 주소를 빼도 126개 사용 가능
    - 첫 번째 주소는 14.24.74.0/25, 마지막 주소는 14.24.74.127/25
  - 두 번째 부-블록에 속하는 주소의 개수는 2의 거듭제곱이 아니기 때문에 64개 주소 할당
    - 첫 번째 주소는 네트워크 주소, 마지막 주소는 특수 목적 주소를 빼도 62개 사용 가능
    - 첫 번째 주소는 14.24.74.128/26, 마지막 주소는 14.24.74.191/26
  - 세 번째 부-블록에 속하는 주소의 개수는 2의 거듭제곱이 아니기 때문에 16개 주소 할당
    - 첫 번째 주소는 네트워크 주소, 마지막 주소는 특수 목적 주소를 빼도 14개 사용 가능
    - 첫 번째 주소는 14.24.74.192/27, 마지막 주소는 14.24.74.207/27

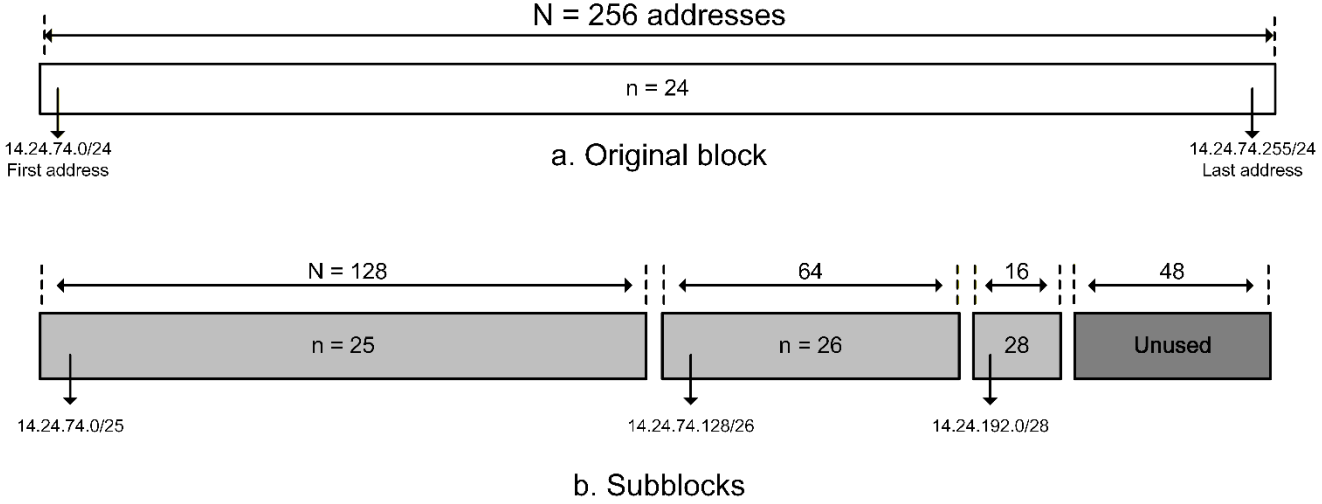
IPv4 주소

- 클래스없는 주소지정
  - 예제 5.25 (2/2)

14.24.74.0/24의 시작 주소를 갖는 하나의 주소 블록이 기관에게 할당되었다. 기관은 다음과 같은 개수의 주소를 갖는 3개의 블록을 구성하고자 한다.

1. 120개의 주소를 갖는 하나의 부-블록
2. 60개의 주소를 갖는 하나의 부-블록
3. 10개의 주소를 갖는 하나의 부-블록

- 풀이
  - 세 개의 부-블록에 할당된 모든 주소를 다 더하면 208
  - 48(256-208)개의 주소가 잔여
  - 첫 번째 주소는 14.24.74.209
  - 마지막 주소는 14.24.74.255
  - 프리픽스 길이는 미지수



IPv4 주소

- 클래스없는 주소지정
  - 예제 5.26

어떤 회사가 중앙, 동쪽, 서쪽의 3개의 사무실을 가지고 있다. 중앙 사무실 사설 WAN 선로를 이용하여 동쪽과 서쪽 사무실과 연결되어 있다. 이 회사가 할당받은 블록은 70.12.100.128/26의 시작주소를 갖는 64개의 주소로 이루어진다. 중앙 사무실에 32개의 주소를 할당하고 나머지 주소를 다른 두 개의 사무실에 할당하라

- 풀이
  - 중앙 사무실 $N_c = 32$, 동쪽 사무실 $N_e = 16$, 서쪽 사무실 $N_w = 16$
  - 각각의 서브 네트워크를 위한 프리픽스 길이는

$$n_c = n + \log_2\left(\frac{64}{32}\right) = 27, \qquad n_e, nw = n + \log_2\left(\frac{64}{16}\right) = 28$$

기호	설명
N_c	중앙 사무실에 할당된 주소의 개수
N_e	동쪽 사무실에 할당된 주소의 개수
N_w	서쪽 사무실에 할당된 주소

기호	설명
N_c	중앙 사무실의 subnetid 길이
N_e	동쪽 사무실의 subnetid 길이
N_w	서쪽 사무실의 subnetid 길이
n	기존 프리픽스의 길이

IPv4 주소

- 클래스없는 주소지정
  - 예제 5.27 (1/2)

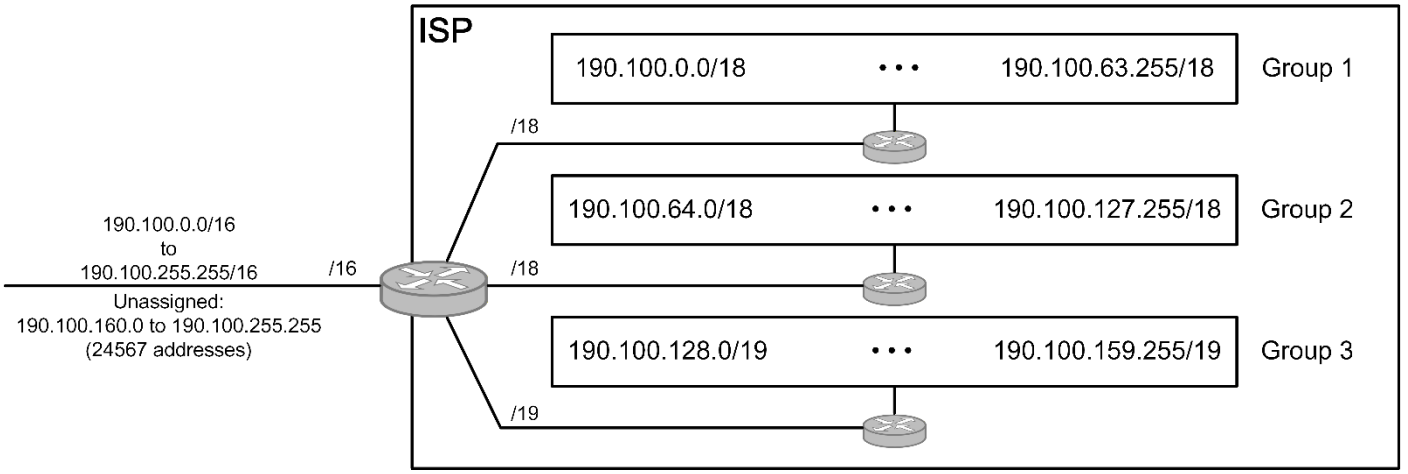
ISP가 190.100.0.0/16(65,356 주소)부터 시작되는 주소 블록을 할당받았다. ISP는 이 주소 블록을 다음과 같은 세 그룹에 고객들에게 배분하고자 한다.

1. 첫 번째 그룹은 64개의 고객을 갖고 있으며, 각각 256개 정도의 주소를 필요로 한다.
2. 두 번째 그룹은 128개의 고객을 갖고 있으며, 각각은 128개 정도의 주소를 필요로 한다.
3. 세 번째 그룹은 128개의 고객을 갖고 있으며, 각각은 64개 정도의 주소를 필요로 한다.

- 풀이
  - 두 단계로 해결
  - 첫 번째 단계
    - 주소의 부-블록을 각각의 그룹에 할당

(각각의 그룹에 할당된 주소의 전체 개수와 각각의 부-블록을 위한 프리픽스 길이)

그룹 1	64 x 256 = 16,384	n1 = 16 + log2(65536/16384) = 18
그룹 2	128 x 128 = 16,384	n1 = 16 + log2(65536/16384) = 18
그룹 3	128 x 164 = 8,192	n3 = 16 + log2(65536/8192) = 19



IPv4 주소

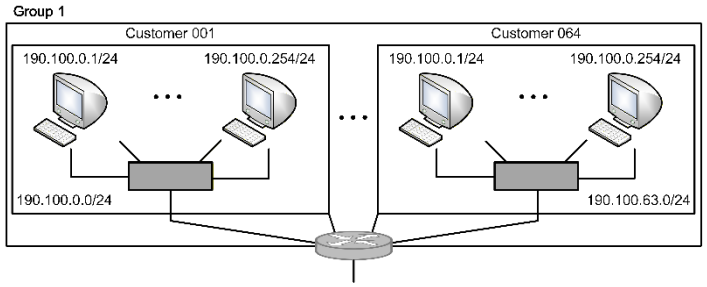
- 클래스없는 주소지정
  - 예제 5.27 (2/2)

ISP가 190.100.0.0/16(65,356 주소)부터 시작되는 주소 블록을 할당받았다. ISP는 이 주소 블록을 다음과 같은 세 그룹에 고객들에게 배분하고자 한다.

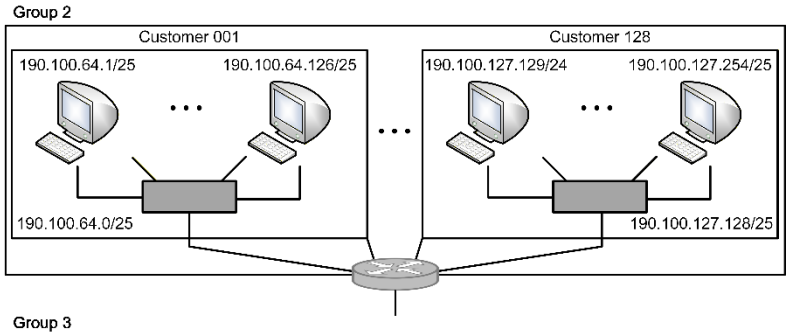
1. 첫 번째 그룹은 64개의 고객을 갖고 있으며, 각각 256개 정도의 주소를 필요로 한다.
2. 두 번째 그룹은 128개의 고객을 갖고 있으며, 각각은 128개 정도의 주소를 필요로 한다.
3. 세 번째 그룹은 128개의 고객을 갖고 있으며, 각각은 64개 정도의 주소를 필요로 한다.

- 풀이
  - 두 단계로 해결
  - 두 번째 단계
    - 각각 그룹 입장에서 생각
    - 그룹 1은 /18, 그룹 2는 /18, 그룹 3은 /19

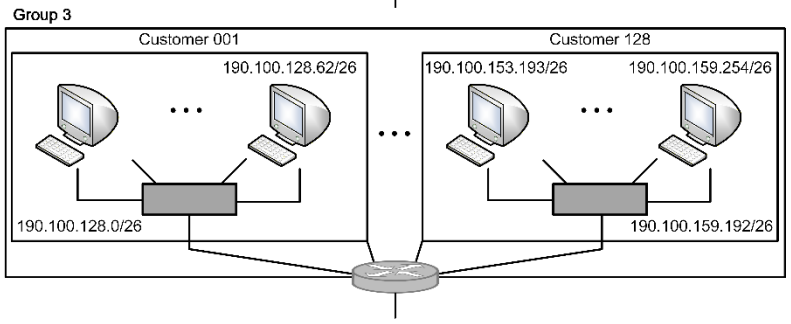
Group: n = 18
Subnet: n = 18 + $\log_2(16384/256) = 24$



Group: n = 18
Subnet: n = 18 + $\log_2(16384/256) = 24$



Group: n = 19
Subnet: n = 19 + $\log_2(8192/256) = 26$

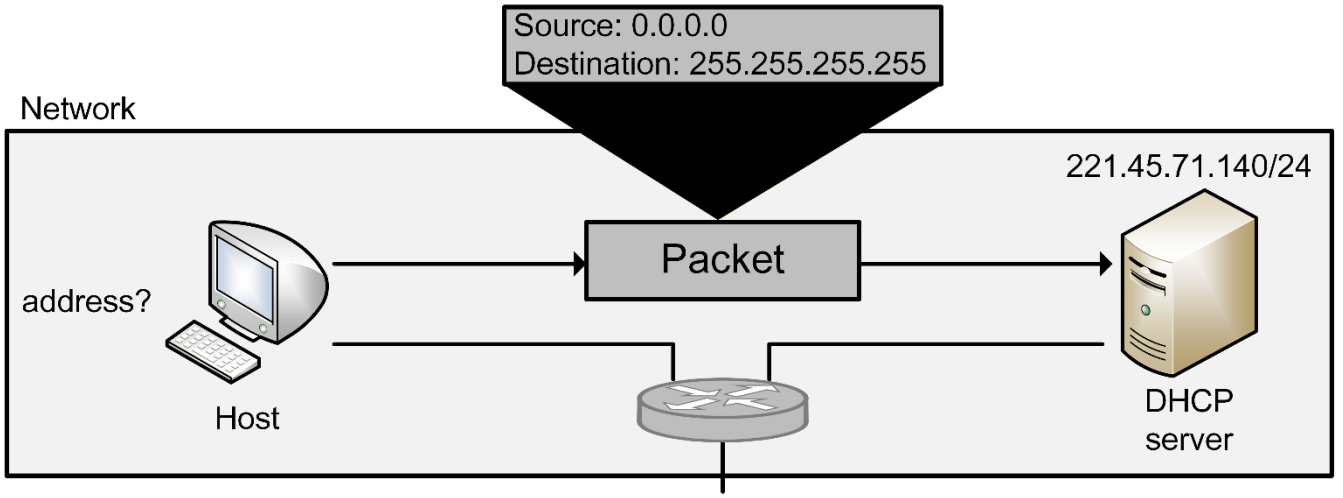


IPv4 주소

- 특수 주소
  - 특수 블록
    - 주소의 일부 블록은 특수 목적을 위하여 예약된 주소

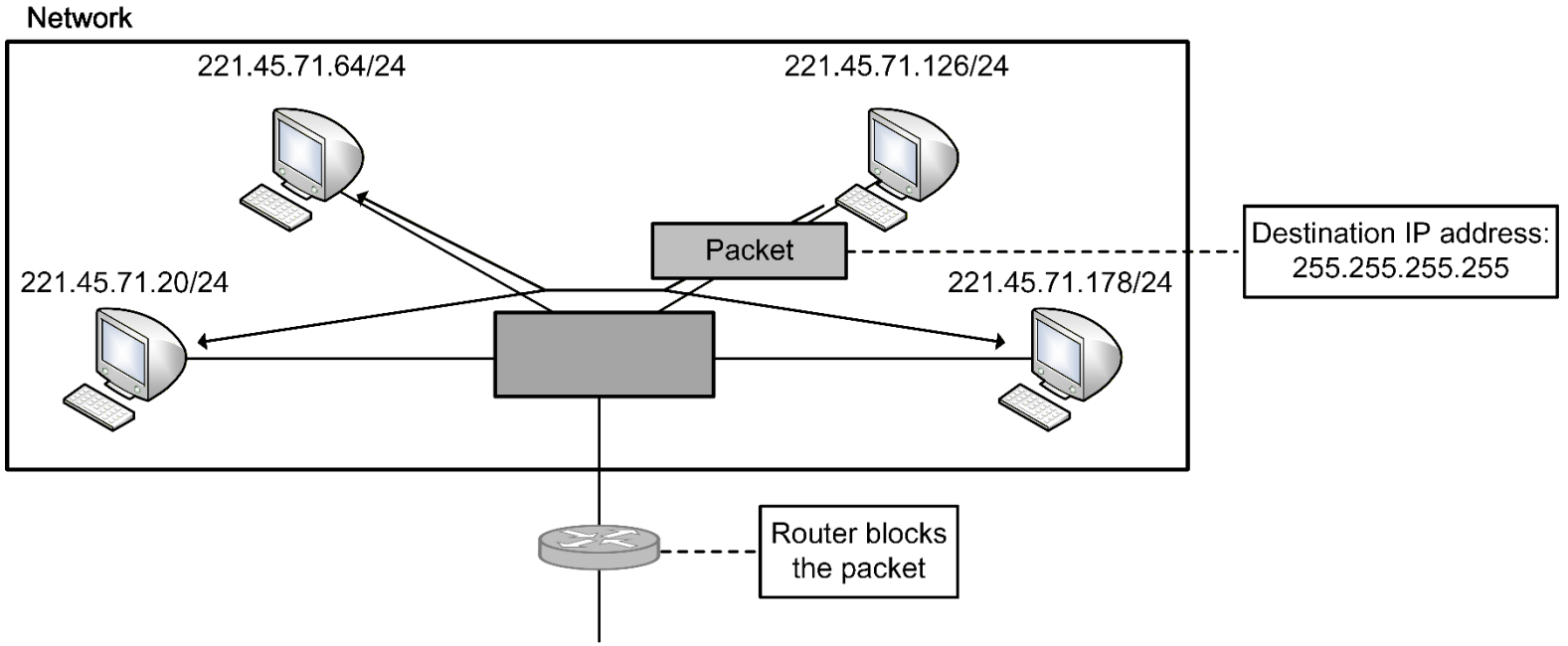
*부트스트랩(Bootstrap)
시스템이나 노드가 네트워크에 처음 연결되어 스스로 초기화하고 통신을 시작하는 것.
*DHCP(Dynamic Host Configuration Protocol)
기기가 네트워크에 연결 될 때, IP 주소, 서브넷 마스크, 기본 게이트웨이, DNS 주소를 자동으로 할당해 주는 프로토콜

    - 종류 (1/4)
      - 모두 0인 주소(0.0.0.0/32)
        - 자신의 IPv4 주소를 모르는 호스트의 IPv4 패킷 전송 시 사용
        - 일반적으로 부트스트랩(Bootstrap)에 사용



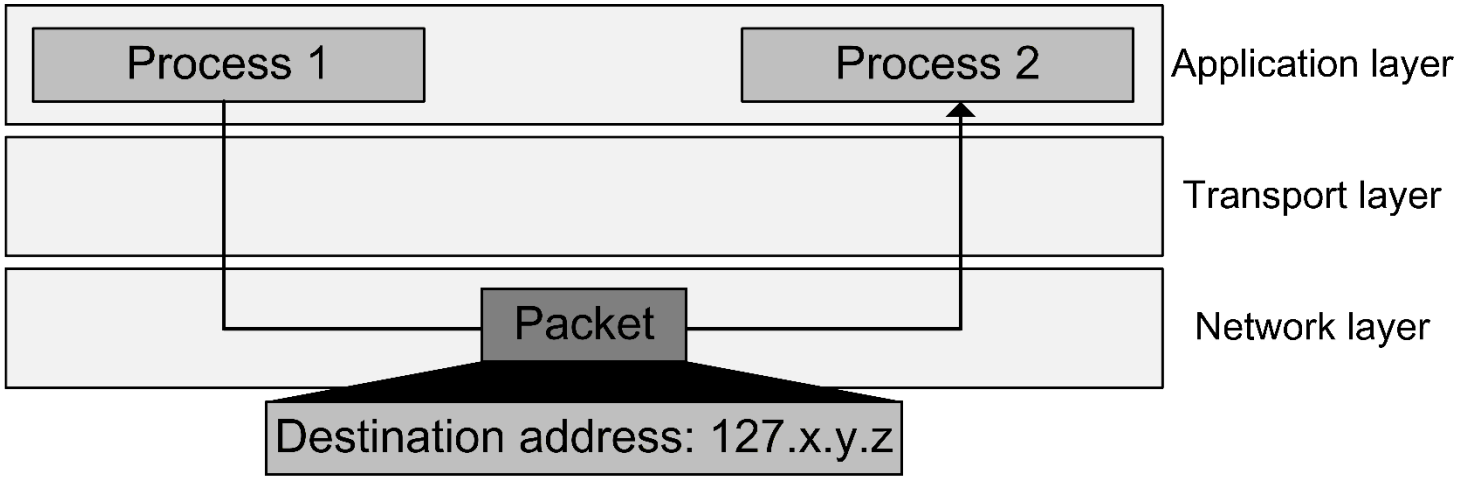
IPv4 주소

- 특수 주소
  - 특수 블록
    - 종류 (2/4)
      - 모두 1인 주소: 제한된 브로드캐스트 주소(255.255.255.255/32)
        - 네트워크 내의 다른 모든 호스트들에게 메시지를 전송할 때 사용
        - 브로드캐스팅을 로컬 네트워크로 한정하기 위한 라우터의 해당 주소 차단



IPv4 주소

- 특수 주소
  - 특수 블록
    - 종류 (3/4)
      - 루프백 주소(127.0.0.0/32)
        - 컴퓨터 설치 소프트웨어 시험용 주소
        - 해당 주소를 사용하는 패킷의 컴퓨터 외부로 전송하지 않고 프로토콜 소프트웨어로 단순 반환
        - IPv4 소프트웨어의 패킷 정상 처리 여부 확인
        - 동일 컴퓨터 내 클라이언트와 서버 프로세스 간 메시지 전송 확인



IPv4 주소

- 특수 주소
  - 특수 블록
    - 종류 (4/4)
      - 사설 주소
        - 사설 용도를 위하여 할당
        - 전역 네트워크에서 인식되지 않음
        - 분리된 네트워크 또는 NAT (Network Address Translation) 이용한 사설 네트워크의 인터넷 연결하는 데 사용

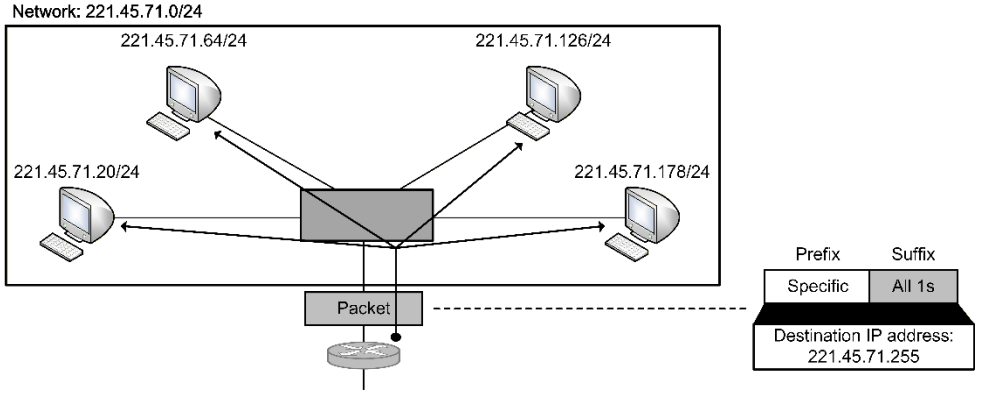
Block	Number of addresses	Block	Number of addresses
10.0.0.0/8	16,777,216	192.168.0.0/16	65,536
172.16.0.0/12	1,047,584	169.254.0.0/16	65,536

          - 10.0.0.0/8: 라우팅 불가능한 사설 IP, 대규모 조직에서 사용
          - 172.16.0.0/12: 라우팅 불가능한 사설 IP, 중간 규모 조직에서 사용
          - 192.168.0.0/16: 라우팅 불가능한 사설 IP, 소규모/가정용 네트워크에서 사용
          - 169.254.0.0/16: 같은 로컬 링크 내에서만 통신 가능하고 라우팅 불가, DHCP 서버로부터 주소를 받았을 때 호스트가 자동으로 자기 자신에게 부여하는 임시 주소
      - 멀티캐스트 주소(224.0.0.0/4)
        - 멀티캐스트 통신을 위하여 예약된 블록

*NAT (Network Address Translation)
하나의 공인 IP 주소를 사용하여 여러 개의 사설 IP 주소를 인터넷에 연결할 수 있도록 해주는 기술.

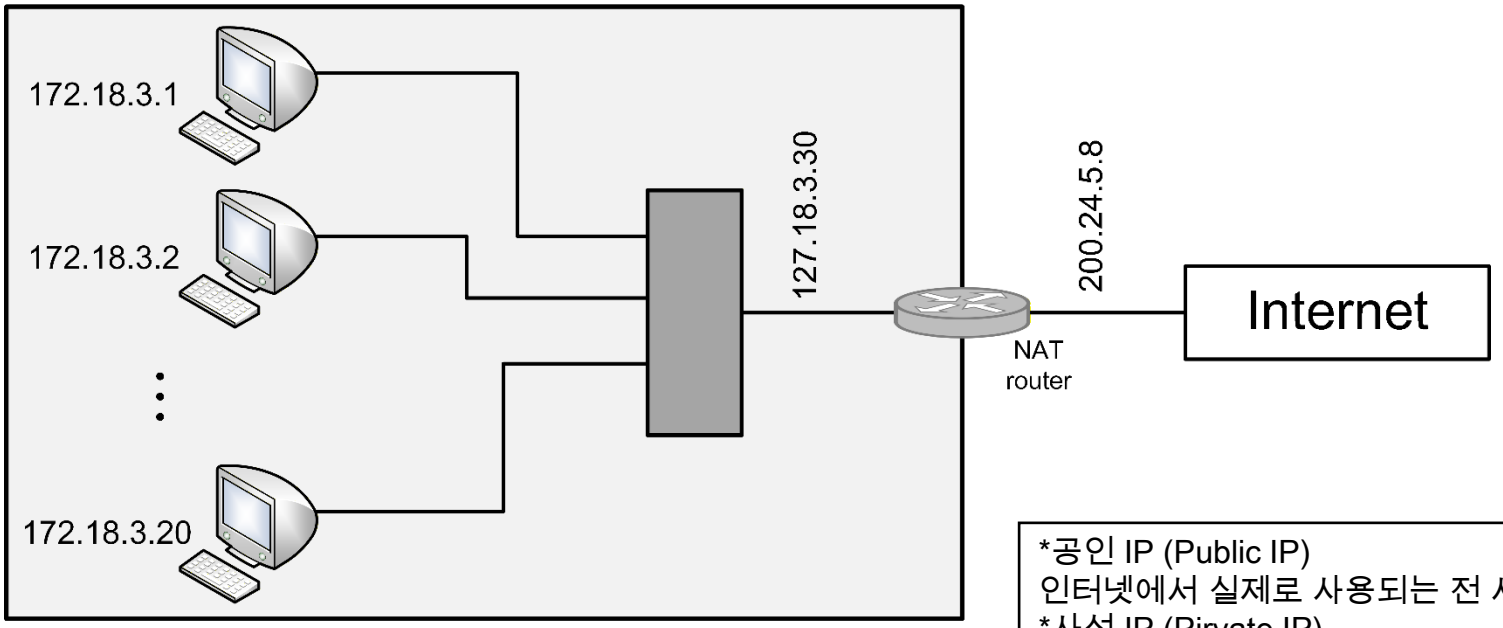
IPv4 주소

- 특수 주소
  - 블록에 속하는 특수 주소
    - 종류
      - 네트워크 주소
        - 블록에 속하는 첫 번째 주소(서픽스(Suffix)는 모두 0으로 설정)
        - 실제로 네트워크에 있는 호스트를 정의하는 것이 아닌 네트워크 자체를 정의
      - 직접 브로드캐스트 주소
        - 서픽스(Suffix)가 모두 1로 설정된 블록 또는 부-블록의 마지막 주소
        - 라우터의 특정 네트워크 내 전체 호스트 대상 패킷 전송하기 위해 사용
        - IPv4 패킷에서 목적지 주소로만 사용 가능



IPv4 주소

- NAT (Network Address Translation) (1/6)
  - 정의
    - 하나의 공인 IP 주소를 사용하여 여러 개의 사설 IP 주소를 인터넷에 연결할 수 있도록 해주는 기술

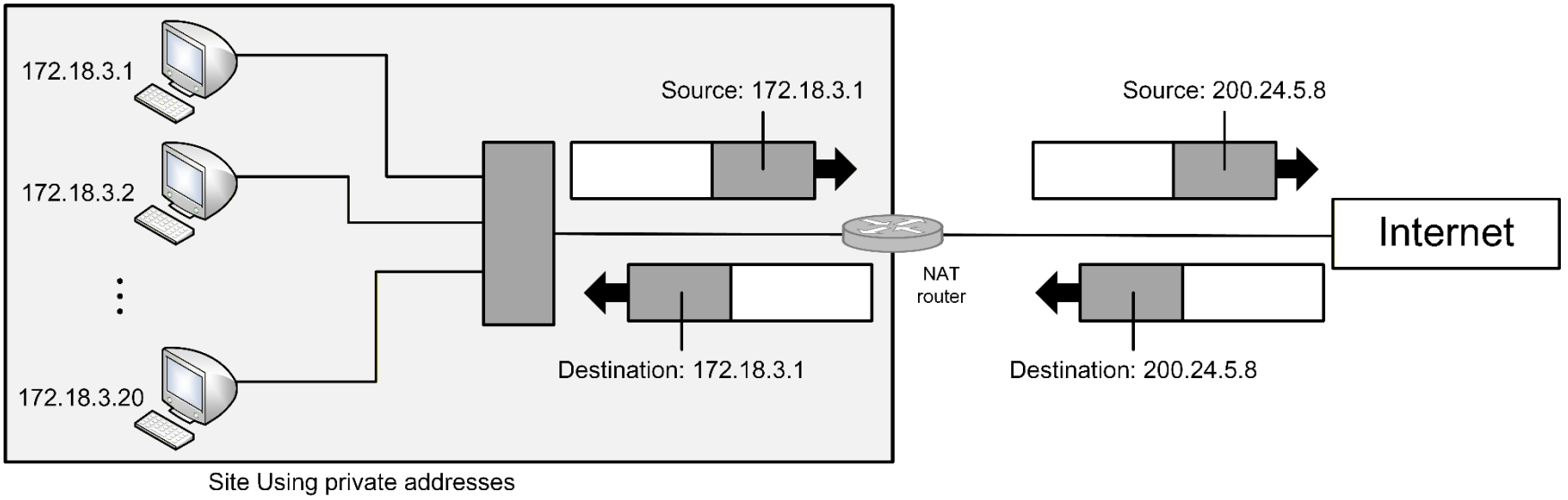


Site Using private addresses

*공인 IP (Public IP)
인터넷에서 실제로 사용되는 전 세계 고유 주소.
*사설 IP (Pirvate IP)
내부 네트워크에서만 사용되는 주소 (외부 인터넷 직접 통신 불가).

IPv4 주소

- NAT (Network Address Translation) (2/6)
  - 특징
    - 주소 변환
      - 외부로 전송되고자 하는 모든 패킷들은 변환 테이블을 사용하여 패킷의 발신지 주소를 전역 주소로 바꾸는 작업

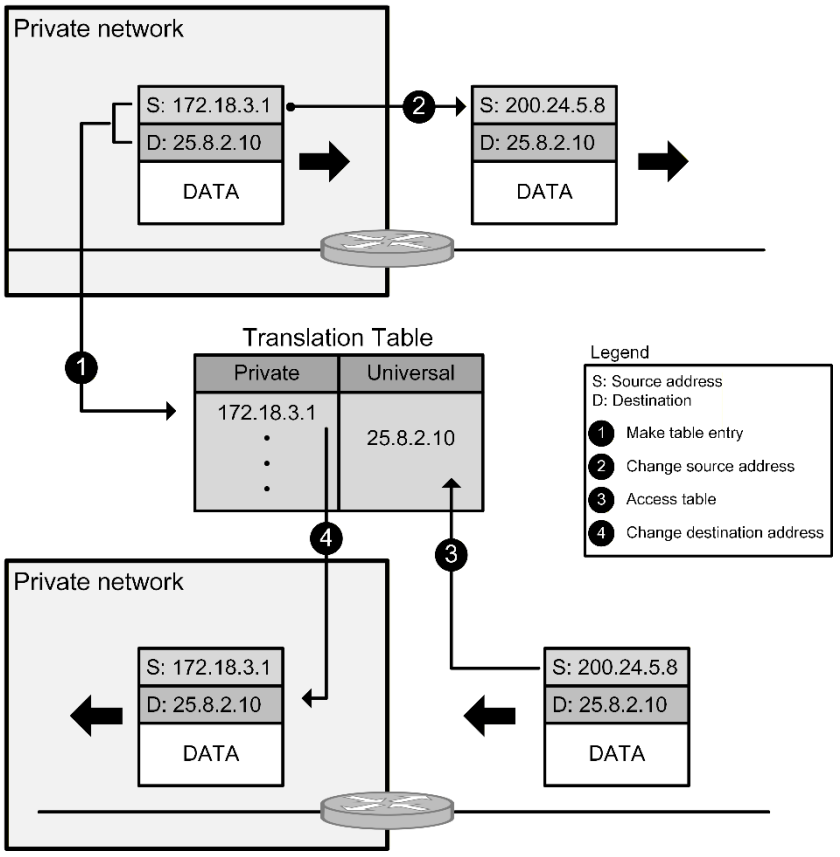


IPv4 주소

- NAT (Network Address Translation) (3/6)
  - 변환 테이블
    - 정의
      - 인터넷으로부터 들어오는 패킷의 목적지 주소를 식별하기 위해 NAT 라우터가 가지고 있는 테이블
    - 종류
      - 하나의 IP 주소를 사용
      - IP 주소 풀(Pool)을 사용
      - IP 주소와 포트 번호 사용

IPv4 주소

- NAT (Network Address Translation) (4/6)
  - 변환 테이블 (1/3)
    - 하나의 IP 주소를 사용
      - 정의
        - 여러 호스트들이 하나의 공인 IP를 사용하는 방식
      - 특징
        - 외부로 나가는 패킷의 발신지 주소를 1개의 공인 주소로 변환
        - 목적지 주소도 함께 테이블에 기록해서, 응답 패킷이 왔을 때 어느 사설 호스트로 돌려줄지 구분
        - 여러 사설 호스트가 존재해도 시간 차이를 두고 같은 공인 주소 1개를 공유
      - 장점
        - 구조가 단순
      - 단점
        - 사설 호스트 1대만 특정 외부 호스트와 통신 가능



IPv4 주소

- NAT (Network Address Translation) (5/6)
  - 변환 테이블 (2/3)
    - IP 주소 풀(Pool) 사용
      - 정의
        - 공인 IP 1개 대신 여러 개(풀)를 묶어 사용하는 방식
      - 특징
        - 사설 호스트가 외부로 나갈 때, 풀에서 비어 있는 공인 IP 1개를 배정
        - "사설 주소 + 배정받은 공인 주소 1개" 조합이 연결 1개로 성립
          - e.g., 공인 IP가 4개면, 서로 다른 사설 호스트 최대 4대까지 동시에 각자 공인 IP를 배정받아 통신 가능
      - 장점
        - 공인 주소 1개일 때의 "사설 호스트 1대만 통신 가능" 제한 해소
      - 단점
        - 동일 외부 목적지로는 풀 크기만큼(4개)만 동시 연결 가능
        - 사설 호스트 1대의 외부 서비스 2개(e.g., HTTP·TELNET) 동시 접속 불가
        - 사설 호스트 2대의 동일 외부 서비스(e.g., HTTP) 동시 접속 불가

IPv4 주소

- NAT (Network Address Translation) (6/6)
  - 변환 테이블 (3/3)
    - IP 주소와 포트 번호 사용
      - 정의
        - 공인 IP에 포트 번호까지 결합하여 사설 호스트 여러 대 ↔ 외부 서버 프로그램 간 다대다 통신을 구분하는 방식
      - 특징
        - 변환 테이블 열을 5개(발신지 주소·포트, 목적지 주소·포트, 전송 계층 프로토콜)로 확장해 모호성 제거
      - 장점
        - 사설 호스트 여러 대 ↔ 외부 서버 프로그램 간 다대다 통신 가능
        - 이전 두 방식의 단점 모두 해소
      - 단점
        - 임시 포트 주소(Ephemeral Port)의 고유성(Uniqueness) 보장 필요

Private Address	Private Port	External Address	External Port	Transport Protocol
172.18.31.1	1400	25.8.3.2	80	TCP
172.18.31.2	1400	25.8.3.2	80	TCP
...	...	...	...	...

Thanks!

서 진 우(jinwoo@pel.sejong.ac.kr)